TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

TUOMAS MÄÄTTÄ

PALVELINTEN VARMISTUSJÄRJESTELMÄT

Diplomityö

Tarkastaja: professori Jarmo Harju
Tarkastaja ja aihe hyväksytty
Tieto- ja sähkötekniikan tiedekunta-neuvoston kokouksessa 8. touko-kuuta 2013

TIIVISTELMÄ

TUOMAS MÄÄTTÄ: Palvelinten varmistusjärjestelmät
Tampereen teknillinen yliopisto
Diplomityö, 61 sivua, 5 liitesivua
Tammikuu 2015
Tietotekniikan diplomi-insinöörin tutkinto-ohjelma
Pääaine: Tietoliikenneverkot ja protokollat
Tarkastaja: professori Jarmo Harju

Avainsanat: Palvelin, varmistus, virtuaalipalvelimet, deduplikointi, tietoturvallisuus, saavutettavuus

Diplomityön tavoitteena oli kuvata yleisellä tasolla niitä seikkoja, jotka vaikuttavat keskikokoisen yrityksen tarpeeseen päivittää oman konesalin palvelinten varmistusjärjestelmä, sekä mitä järjestelmän uusinnassa tulisi ottaa huomioon. Näiden yhteydessä työssä pureudutaan myös jonkin verran itse konesalitekniikkaan, varsinkin niiden teknologioiden osalta, jotka eivät kuulu tavallisen toimiston it-infrastruktuuriin.

Diplomityössä käsitellään ensisijaisesti varmistusjärjestelmän käyttöä omassa konesaliympäristössä. Kun it-infrastruktuuri ja varmistusjärjestelmä ovat kokonaan omassa ylläpidossa, on myös sen määrittelyihin helpompi vaikuttaa. Pilvipalvelun hyödyntäminen osana palautumissuunnitelmaa saattaa olla organisaatiolle kustannustehokas ratkaisu, mutta sen käyttö tulisi ehdottomasti testata huolellisesti.

Työssä on pyritty korostamaan myös varmistuksiin liittyvää hallinnollista puolta, sillä tarve datan varmistamiselle, eli datan saavutettavuuden turvaamiselle, tulisi olla määritetty organisaation tietoturvastrategiassa.

Päivittäisten varmistusten ottaminen deduplikoivalle levypohjaiselle järjestelmälle ja viikoittaisten ottaminen nauhoille osoittautui toimivaksi järjestelyksi suurimmalle osalle varmistustöistä. Tällöin tuotannon kannalta kriittisimmistä palvelimista on varmistukset hajautettu kahdelle eri medialle ja levypohjaiselta järjestelmältä yksittäisten tiedostojen palauttaminen on monissa tapauksissa nopeampaa kuin nauhakirjastosta. Tällöin myös mahdollinen deduplikointikannan korruptoituminen ei aiheuta kohtuuttoman suurta datan häviämisen riskiä.

Diplomityön lopussa selvitettiin myös uuden järjestelmän suorituskykyä ja sen resurssien riittävyyttä. Deduplikointikannan levytilan, varmistuspalvelimen prosessorin ja muistin kapasiteetti riittäisi vielä merkittävästi suuremman datamäärän käsittelyyn. Sen sijaan varmistettavan datan siirtämiseen käytetty verkkoyhteys vaikuttaisi olevan järjestelmän pullonkaula, eli varmistustöiden kesto näyttäisi riippuvan pitkälti verkkoyhteyden nopeudesta.

ABSTRACT

TUOMAS MÄÄTTÄ: Backup systems for servers
Tampere University of Technology
Master of Science Thesis, 61 pages, 5 Appendix pages
January 2015
Master's Degree Programme in Information Technology
Major: Communication Networks and Protocols
Examiner: Professor Jarmo Harju

Keywords: Server, backup, virtual server, deduplication, information security, availability

The objective of this Master's thesis is to describe broadly those factors that affect medium sized enterprises' needs to update their backup systems in their data center and also some details one should take into account. On the other hand this theses delves a bit into technology used in the data centers, especially those that are not part of conventional office IT infrastructure.

This thesis focuses on backup systems that are used to backup data from a self-maintained data center. It is easier to control backups when IT infrastructure and backup systems are not outsourced. Cloud based disaster recovery site as a part of disaster recovery plan may be very cost-effective but it must be tested extensively.

Administrative role of backups is also highlighted in this thesis. The need of taking backups, in other words, maintaining data availability, should be defined in the organization's information security strategy.

Daily backups to deduplicated disk based storage and weekly backups to tape library turned out to be practical solution for most of the servers that need to be backed up. In this way servers that have the most critical data have backups distributed into two different type of media. In the unlikely event of corrupted deduplication database there is still data to be restored from an alternative source. In addition, restoring files from disk based backup storage is in many cases much faster than from a tape library.

Performance analysis and resource evaluation of the new backup system has been carried out at the end of this thesis. The size of the disk based storage for the deduplication database, processor performance and memory capacity was found to be sufficient for even larger amount of data to be processed. Instead, network connection seems to be the bottle neck for the backup system. This means that backup job durations depend highly on the bandwidth of the network connection.

ALKUSANAT

Kiitokset tämän diplomityön mahdollistamisesta työnantajalleni Trimico Oy:lle sekä emoyhtiölle Oy Matkahuolto Ab:lle, joka toimi tämän työn osalta asiakkaana toteutuneessa projektissa. Erityinen kiitos kuuluu myös diplomityön tarkastajalle, professori Jarmo Harjulle; diplomityön ohjaajalle, kehityspäällikkö Juha Sahalalle sekä omalle lähimmälle esimiehelleni ja työtovereilleni.

Itse diplomityö syntyi asiakkaan tilaaman projektin yhteydessä, kun jo useamman vuoden käytössä ollut järjestelmä päätettiin korvata uudemmalla. Otin tämän pienimuotoisen projektin ilolla vastaan työantajaltani. Itse projektiin kuului muun muassa tarpeiden kartoitusta, vaatimusten määrittelyä ja vaihtoehtoisten järjestelmien ominaisuuksiin sekä kustannuksiin perehtymistä. Diplomityö keskittyy kuitenkin järjestelmän tekniseen tarkasteluun hankinnan sijasta. Järjestelmän varsinainen asennus toteutettiin yhteistyössä ulkopuolisen konsultin kanssa.

Helsingissä, 15.12.2014

Tuomas Määttä

SISÄLLYSLUETTELO

LYHENTEET

ANSI	American National Standards Institute
CCS	Common Command Set, SCSI-käskyjä
CP	Contingency Planning, valmiussuunnitelma
DC	Domain Controller, toimialueen hallintapalvelin
DHCP	Dynamic Host Configuration Protocol, verkkoasetusten määrittelyyn käytetty protokolla
DNS	Domain Name System, nimipalvelujärjestelmä
DOS	Denial of Service, palvelunestohyökkäys
DRP	Disaster Recovery Plan, palautumissuunnitelma
DRS	Distributed Resource Scheduler, hajautettujen resurssien vuoronnus
DSL	Digital Subscriber Line, digitaalinen tilaajayhteys
D2T	Disk-to-Tape, vaiheistus levyltä nauhalle
D2D	Disk-to-Disk, vaiheistus levyltä levylle
D2D2T	Disk-to-Disk-to-Tape, vaiheistus levylle ja nauhalle
D2C	Disk-to-Cloud, vaiheistus levyltä pilveen
D2D2C	Disk-to-Disk-to-Cloud, vaiheistus levylle ja pilveen
EDI	Electronic Data Interchange, sähköinen tiedonsiirto
ENISA	European Network and Information Security Agency
FC	Fibre Channel, kuitukanava
FCIP	Fibre Channel over IP, IP-verkkoa käyttävä kuitukanava
FC-AL	Fibre Channel Arbitraded Loop, rengastopologiaa käyttävä kuitukanava
FC-SW	Fibre Channel Switched Fabric, kytketty kuitukanava
FIPS	The Federal Information Processing Standards
HBA	Host Bus Adapter, isäntäväyläsovitin
INCITS	InterNational Committee on Information Technology Standards
IOPS	Input/Output Operations Per Second, kirjoitus- ja lukuoperaatioiden lukumäärä sekunnissa
iSCSI	Internet Small Computer System Interface, TCP/IP-verkkoa käyttävä oheislaiteväylä
ISL	Inter-Switch Link, kytkinten välinen linkki
LTO	Linear Tape Open, avoin nauhamediastandardi
MPLS	Multiprotocol Label Switching, yritysverkkojen toteuttamiseen käytetty protokolla
MTU	Maximum Transmission Unit, Ethernet-kehyksen suurin sallittu koko
NAS	Network Attached Storage, verkkotallennusjärjestelmä
NFS	Network File System, verkkotiedostojärjestelmä
NIC	Network Interface Controller, verkkosovitin
NIST	National Institure of Standards and Technology
OSI	Open Systems Interconnection
RAID	Redundant Array of Independent Disks, usean levyn yhdistäminen paremman vikasietoisuuden tai suorituskyvyn saavuttamiseksi
RAM	Random Access Memory, keskusmuisti
RCV WND	Receiver Window, TCP-protokollan vastaanottoikkunan koko
ROI	Return on Investment, sijoitetun pääoman tuotto

ROSI	Return on Security Investment, tietoturvallisuuteen sijoitetun pääoman tuotto
RPO	Recovery Point Objective, palautumispiste
RTO	Recovery Time Objective, toipumiskyky
RTT	Round Trip Time, päästä−päähän-viive
SAN	Storage Area Network, tallennusverkko
SAS	Serial Attached SCSI, sarjaliitäntäinen SCSI
SCSI	Small Computer System Interface, oheislaiteväylä
SLA	Service Level Agreement, palvelutasosopimus
SNIA	The Storage Networking Industry Association
SSD	Solid State Drive, puolijohdelevy
TCP/IP	Transmission Control Protocol/Internet Protocol, protokollaperhe
VAHTI	Valtionhallinnon tieto- ja kyberturvallisuuden johtoryhmä
VPN	Virtual Private Network, virtuaalinen yksityisverkko
VSS	Volume Shadow Copy Service, käytössä olevien tiedostojen avaamiseen tarkoitettu tekniikka
WAN	Wide Area Network, laajaverkko
WWN	World Wide Name, maailmanlaajuinen nimi
WWNN	World Wide Node Name, maailmanlaajuinen noodinimi
WWPN	World Wide Port Number, maailmanlaajuinen porttinumero

1. JOHDANTO

Tiedon merkitys nyky-yhteiskunnassa on suunnaton. Suomen elinkeinoelämässä on valmistavan teollisuuden merkitys pienentynyt. Tiedon merkitys yhtenä tuotannontekijänä onkin valtava, silloin kun yritys ei valmista mitään käsin koskettavia tuotteita. Näin on etenkin asiakasyrityksessä, jonka järjestelmiä tässä diplomityössä käsitellään. Liiketoiminta perustuu palveluiden myymiseen loppuasiakkaille. Tässä tapauksessa palveluilla tarkoitetaan rahdin ja ihmisten kuljettamista paikasta toiseen. Kumpaakin palvelua on tarjottu ennen kuin tietotekniikkaa osattiin laajemmin hyödyntää liiketoiminnassa, mutta tietotekniikka mahdollistaa näiden toteuttamisen kustannustehokkaasti sekä tarjoten asiakkaille monipuolisempia palveluita, kuten esimerkiksi lähetysten toimitusseurannan, bussien aikatauluhaut sekä rahtiluottoasiakkaiden lähetysasiakirjojen tallentamisen selainkäyttöliittymän avulla tai jopa suora yritysten välinen EDI-tiedonsiirto. Lisäksi tietotekniikan hyödyntäminen on lähes itsestäänselvyys yritysten taloushallinnossa ja muissa toiminnan ohjaukseen ja päätöksentekoon liittyvissä toimissa. Tiedonhallinnan alalla yksi tuoreista trendeistä on big data, jolla tarkoitetaan suurten jäsentymättömien tietomäärien käyttöä (LVM 2013). Tiedon hallinta on nykyään kiistämättä osa yritysten kilpailukykyä, mutta sen merkitystä ei sovi vähätellä julkisellakaan puolella.

Tässä diplomityössä keskitytään yrityksen toiminnan kannalta kriittisen tiedon turvaamiseen, tarkemmin sanottuna tiedon saatavuuden varmistamiseen varmuuskopioinnin menetelmin. Datan varmuuskopioiminen on monitahoinen toimenpide, johon vaikuttaa niin hallinnolliset kuin myös tekniset seikat. Tässä diplomityössä pääpaino on teknisellä puolella. Huomioitavaa on, että alan suomenkielinen tekninen käsitteistö ei ole täysin vakiintunutta, joten työssä on myös pyritty riittävän tarkasti selittämään termien merkitys, jotta lukijalle ei pääse syntymään väärinkäsityksiä. Käsitteistä ilmaistaan myös niiden englanninkielinen vastine.

Varmistusjärjestelmistä tai niihin läheisesti liittyen ei ole suomalaisissa yliopistoissa kovinkaan paljon aikaisempia diplomitöitä kirjoitettu. Hokkanen (2007) esitti joukon tietoturvallisuuteen kohdistuvia haasteita tietoturvallisuuteen liittyen sekä näihin mahdolliset parannukset, kun Linux-palvelinten tietoja varmuuskopioidaan palvelimelta toiselle vapaita ohjelmia käyttäen. Tässä diplomityössä tietoturvallisuus informaation luottamuksellisuuden kannalta ei ole keskiössä, vaan tarkoituksena on luoda enemmänkin katsaus palvelinten varmistusjärjestelmiin yleisemmin.

1.1 Lähtökohta

Lähtökohta tämän diplomityön kirjoittamiseen tuli asiakasyrityksen tarpeesta päivittää olemassa oleva varmuuskopiointijärjestelmä uuteen, sillä vanha järjestelmä oli tulossa käyttöikänsä puolesta loppuun eikä järjestelmä toisaalta enää vastannut kasvaneita ja muuttuneita vaatimuksia. Varmistusjärjestelmät ovat tyypillisesti hyvin tiiviisti tietojärjestelmien infrastruktuuriin integroituvia, joten tämän infrastruktuurin uudistuminen saattaa asettaa myös uusia vaatimuksia varmistusjärjestelmää kohtaan. Varmistusjärjestelmän päivittämisellä valmistauduttiin myös myöhemmin tehtävän virtuaaliympäristön uudistamiseen. Toisaalta uusi varmistusjärjestelmä ei välttämättä enää tue vanhempia järjestelmiä. Tietojärjestelmissä olevan datamäärän kasvu oli myös syynä päivittää varmistusjärjestelmä, sillä vanha oli kapasiteettinsa rajoilla. Järjestelmän sisältäessä mekaanisia osia, kasvaa myös ajan myötä niiden vikaantumisen todennäköisyys, joten koko järjestelmän uudistamisella ennaltaehkäistiin ongelmien syntymistä.

Asiakkaana toimiva yritys omistaa itse suurimman osan päivittäisestä tietotekniikkainfrastruktuuristaan, jota puolestaan ylläpitää ja kehittää tytäryhtiö. Tietotekniset ratkaisut ovat siis varsin pitkälle yhtiön omissa käsissään ja yhtiö on niiden osalta kohtalaisen omavarainen.

1.2 Tutkimusongelma

Tämä diplomityö pyrkii vastaamaan kahteen erilliseen tutkimusongelmaan, jotka liittyvät varsin tiiviisti toisiinsa. Ensin pohditaan, että mitkä asiat johtavat tarpeeseen päivittää käytössä oleva varmistusjärjestelmä. Tähän pureudutaan tämän diplomityön luvussa 4.3. Toisena asiana ja selkeänä jatkokysymyksenä selvitetään, että mitkä ovat keskeisimpiä huomioitavia asioita varmistusjärjestelmän suunnittelussa. Näitä tekijöitä on pohdittu luvussa 5.

Varmistusjärjestelmän päivittämisellä tarkoitetaan tässä laajasti kaikkia järjestelmään tehtäviä olennaisia muutoksia ja ohjelmiston versiopäivityksiä. Järjestelmän päivittämiseen johtavat tarpeet voivat johtua sekä teknisistä että hallinnollisista seikoista. Tässä tapauksessa asiakkaan varmistusjärjestelmä koki kuitenkin varsin perusteellisen muutoksen, sillä uusittavaksi meni sekä itse laitteisto että ohjelmisto. Käyttöön otettiin myös alkuperäisestä järjestelmästä jonkin verran poikkeavia, nykyaikaisempia tekniikoita, joista merkittävin lienee luvussa 3.7 esitelty datan deduplikointi.

Toisen tutkimusongelman osalta tarkastellaan itse varmistusjärjestelmän suunnittelun kannalta keskeisiä huomioitavia asioita. Erityisesti pyritään pohtimaan, että miten järjestelmä skaalautuu varmistettavan ympäristön laajentuessa, jolla ensisijaisesti tarkoitetaan varmistettavan datamäärän kasvua, mutta myös ympäristön infrastruktuuria koskevia muutoksia sekä kokonaan uudentyyppisiä varmistettavia järjestelmiä.

1.3 Tutkimuksen rajaus

Tässä diplomityössä tarkastellaan vain palvelinten datan varmistusta ja täten diplomityön aihealueen ulkopuolelle rajataan tavallisten työasemien varmistus. Asiakkaan työasemaympäristö on suunniteltu siten, että itse työasemien datan varmistamista ei muutamia poikkeustapauksia lukuun ottamatta tarvita, sillä työasemilla sijaitseva data sijaitsee pääasiassa verkkolevyillä tai kokonaan erillisissä tietojärjestelmissä. Tämän ansiosta vikaantunut työasema voidaan korvata joko väliaikaisesti tai kokonaan uudella vastaavalla vakioidulla työasemalla, ilman että käyttäjien tai ylläpidon tarvitsee huolehtia datan palauttamisesta. Myöskään asiakkaan käytössä oleva verkkoinfrastruktuuri ei tue työasemien keskitettyä varmuuskopiointia.

Diplomityö myös pyrkii tekemän selvän eron varmuuskopioinnin, replikoinnin ja arkistoinnin välille. Kaikki nämä kolme datan tallentamiseen liittyvää menetelmää ovat kuitenkin jossain muodossa käytössä asiakkaan tietojärjestelmissä.

1.4 Käytetyt menetelmät ja aineisto

Tämä diplomityö on kirjoitettu osana tuotantoympäristössä toimivan varmistusjärjestelmän uudistamisprojektia ja työhön on myös kerätty käytännön kokemuksia varmistusjärjestelmän toiminnasta, joista suurin osa koskee uutta käyttöön otettua järjestelmää. Havaintojakson pituus on noin puolitoista vuotta, jonka aikana järjestelmän kyvyt ja puutteet käytetyssä ympäristössä tulivat varsin selkeästi esille.

Työssä esitetyt mittaustulokset ja tilastot ovat otettu todellisesta tuotantoympäristöstä, jossa kaikkien muuttujien vaikutusta ei ole mahdollista eliminoida. Tämä asettaa erityisiä haasteita mahdollisten virhelähteiden arviointiin. Vastaavat mittaukset ovat kuitenkin toteutettavissa lähes missä tahansa samankaltaisessa järjestelmässä, sillä mittaamiseen käytetyt työkalut ovat vapaasti saatavilla tai valmiiksi osa käyttöjärjestelmää. Tilastot varmuuskopioiden kestoista ja tallennetuista datamääristä ovat suoraan varmistusjärjestelmän käyttämästä sovelluksesta. Bittien ja tavujen muutoksessa on käytetty kymmenkantaista järjestelmää kaikissa laskutoimituksissa.

Virhelähteiden vaikean arvioimisen takia työssä pohditaan varmistusten merkitystä ja vaikutusta myös yleisemmällä tasolla. Tämän yhteydessä on työssä esitettyjä päätelmiä tuettu kirjallisuuslähteillä. Mittaustulosten esittämistä pohjustetaan avaamalla käytössä olevaa konesalitekniikkaa sekä varmistusjärjestelmään sisältyviä käsitteitä. Lähteinä tässä on käytössä esimerkiksi laite- ja ohjelmistovalmistajien teknisiä oppaita tieteellisten artikkelien lisäksi.

1.5 Diplomityön rakenne

Johdannon jälkeisessä toisessa luvussa pohjustetaan tulevaa pohtimalla varmistusten roolia yleisesti yhtenä tietoturvallisuuden osa-alueena sekä sen vaikutusta yrityksen liiketalouden kannalta. Tässä luvussa tarkastellaan palautumissuunnitelman (DRP, Disaster Recovery Plan) merkitystä sekä siihen liittyvien mitattavien suureiden, palautumispisteen ja toipumiskyvyn, tarkoitusta. Lisäksi selvitetään ero tiedon arkistoinnin ja varmuuskopioinnin osalta sekä tutustutaan muutamiin alalla julkaistuihin ohjeisiin ja standardeihin. Toinen luku sisältää siis pääosin palvelinten varmistusten hallinnollista puolta.

Kolmas luku on selkeästi teknisempi ja sisältää katsauksen teknisiin ratkaisuihin, joita on asiakkaan uudessa käyttöönotetussa varmistusjärjestelmässä. Näiden lisäksi selvitetään muutamia laitesalien toiminnan kannalta merkittävimpiä tekniikoita joita ovat esimerkiksi levyjärjestelmät, SAN-verkot ja virtuaalipalvelimet.

Neljännessä luvussa kuvataan lähtötilanne, eli vanha varmistusjärjestelmä sekä siinä havaitut haasteet. Tämän lisäksi esitetään asiakkaan käytössä oleva palvelinsaliympäristö sekä tietoliikenneratkaisut pääpiirteissään, jotta lukijalle välittyy kuva järjestelmän kokonaistilanteesta. Yksityiskohtaista tietoa järjestelmästä ei kuitenkaan kerrota tietoturvallisuussyistä.

Viidennessä luvussa kuvataan uusi varmistusjärjestelmä ja erityisesti merkittävimmät erot vanhaan järjestelmään. Luvussa tulee ilmi myös implementoinnin aikana koetut haasteet sekä niiden syitä. Kuudes luku sisältää kvantitatiivista tutkimusta vanhan ja uuden järjestelmän suorituskykyjen eroista, joilla pyritään tuomaan selvästi esille erilaisten tekniikoiden väliset erot. Toisaalta tarkoituksena on myös tuoda ilmi uuden järjestelmän parantunut suorituskyky numeroiden valossa.

Diplomityön päättää seitsemäs luku, johon kootaan yhteenveto koko työstä. Luvussa myös kerrotaan uudella varmistusjärjestelmällä saavutetut hyödyt ja toisaalta järjestelmän haasteet sekä pyritään luomaan eräänlainen lista toimiviksi koetuista parhaista käytänteistä.

2. VARMISTUSTEN ROOLI TIETOTURVALLISUUDESSA

Tiedon varmistaminen ei ole puhtaasti tekninen operaatio, vaan itse tiedon tulisi olla keskeisessä asemassa esimerkiksi määritettäessä palvelinten varmistuksia tai luotaessa organisaation palautumissuunnitelmaa. Tässä luvussa käsitellään tiedon varmistamista aluksi hieman teoreettisemmalta näkökulmalta, eli osoitetaan tämän rooli osana tietoturvallisuutta. Tämän lisäksi esitetään myös varmistuksiin liittyvää hallinnollista puolta.

2.1 Yleistä tietoturvallisuudesta

Tietoturvallisuuteen liitetään yleisesti neljä osa-aluetta, jotka ovat luottamuksellisuus (engl. confidentiality), eheys (engl. integrity), saatavuus (engl. availability) ja kiistämättömyys (engl. non-repudiation). Luottamuksellisuudella tarkoitetaan, että data on vain valtuutettujen toimijoiden käytettävissä, eikä paljastu tahallisesti tai tahattomasti ulkopuolisille. Eheys toteutuu, mikäli dataan ei pääse ulkopuoliset tekemään muutoksia, lisäyksiä tai poistoja. Eheys ei kuitenkaan takaa, eikä myöskään vaadi, tiedon oikeellisuutta. Saatavuudella tarkoitetaan, että valtuutetuilla toimijoilla on mahdollisuus käsitellä dataa tarvittaessa. Kiistämättömyys on sitä, että toimija ei pysty jälkikäteen kieltämään datan saapumista tai vastaavasti väittämään vastaanotetun datan saapumattomuutta. (Cleveland 2008)

Varmuuskopioinnilla pyritään varmistamaan datan saatavuus myös silloin, kun tietojärjestelmästä data on hävinnyt esimerkiksi teknisen vian tai inhimillisen erehdyksen vuoksi. Täten on ilmeistä, että neljästä edellä mainitusta osa-alueesta juuri saatavuus liittyy kiinteästi datan varmistamiseen.

Väheksyä ei kuitenkaan sovi luottamuksellisuuden ja eheyden merkitystä. Tietojärjestelmissä oleva tieto saattaa olla hyvinkin arkaluonteista ja sen turvaamiseen on saatettu käyttää merkittävästi resursseja. Yhtä suuri huomio tulisi tällöin kohdistaa varmuuskopioiden luottamuksellisuuden turvaamiseen, sillä muutoin järjestelmässä saattaa hyökkääjälle olla poikkeuksellisen suora tie arkaluonteiseen tietoon. Esimerkiksi nauhavarmuuskopiot saattavat päätyä asiaan kuulumattomien käsiin kuljetettaessa paikasta toiseen. Nauhoilla olevan datan salaaminen saattaa estää nauhoilla olevan datan päätymisen luvattomiin käsiin, mutta tämä luo lisää samalla hieman lisähaasteita salausavainten hallintaan. Samalla tulee varmistua, että data pystytään palauttamaan tarvittaessa myös muulla nauhurilla, kuin millä varmuuskopio on nauhalle kirjoitettu. Varmistuksilla ei

myöskään ole merkitystä, mikäli data ei ole eheää, eli vastaa tarkalleen alkuperäistä. Tämän takia tulisi varmistustöiden yhteydessä myös varmistaa datan eheys.

2.2 Kustannukset

European Network and Information Security Agency (ENISA) on Euroopan unionin alainen tutkimuskeskus, jonka tarkoituksena on auttaa sen jäseniä tunnistamaan, vastaamaan niihin sekä erityisesti ennalta ehkäisemään tietoturvallisuusongelmia. Organisaatioiden ei ole helppoa laskea tietoturvallisuuteen sijoitetuille rahoille tuottoa ja siten arvioida niiden kannattavuutta, sillä turvallisuus ei yleensä ole tuottava investointi, vaan kyseessä on lähinnä häviöiden estämiseen liittyvä investointi. ENISA on esittänyt talouden tunnusluvuista tutusta sijoitetun pääoman tuottoprosentista (Return on Investment, ROI) jalostetun version, tietoturvallisuuteen sijoitetun pääoman tuottoprosentin (Return on Security Investment, ROSI). Yksinkertaisesti ROSI-prosentti voidaan laskea seuraavasti:

$$ROSI = \frac{Rahallisten\ häviöiden\ vähentyminen - ratkaisun\ kustannukset}{ratkaisun\ kustannukset} \quad (2.1)$$

Mikäli tällä lasketulla kaavalla saadaan tulokseksi suurempi kuin yksi, eli ROSI-prosentiksi yli sata, voidaan investointia pitää kannattavana. Laskennassa voidaan käyttää apuna lisäksi seuraavia suureita:

- SLE (Single Loss Expectancy): Yksittäisen insidentin aiheuttama rahallinen häviö
- ARO (Annual Rate of Occurance): Insidenttien todennäköinen vuotuinen määrä
- ALE (Annual Loss Expectancy): SLE:n ja ARO:n tulo, odotettavissa olevat vuotuiset häviöt
- mALE (modified ALE): Odotettavissa olevat vuotuiset häviöt, kun insidentin toteutumista tai vaikutusta pyritään estämään tietoturvasovelluksella

Käyttäen näitä suureita, voidaan ROSI-prosentti laskea seuraavasti:

$$ROSI = \frac{ALE - mALE - ratkaisun\ kustannukset}{ratkaisun\ kustannukset} \quad (2.2)$$

Yhtälön suureista ratkaisun kustannukset ovat helpommin ennustettavissa, mutta muilta osin suureet ovat arvioita, joten tietoturvallisuuteen sijoitetun pääoman tuottoprosentti on myös vain arvio ja arvioinnissa tulisi käyttää edellisien vuosien aikana organisaation kohdistuneita tietoturvapoikkeamia myyjien esittämien tutkimusten sijasta. (ENISA 2012)

Varmuuskopioiden ottaminen tietojärjestelmien sisältämästä datasta ei ole ainoa ROSI:in kuuluva asia, mutta osana suurempaa tietoturvallisuuskokonaisuutta, varmuuskopiot liittyvät olennaisesti tietoturvallisuuteen sijoitetun pääoman tuottoprosentin laske-

miseen. Mallin käyttökelpoisuutta rajoittaa sen epätarkkuus ja toisaalta arvioinnin vaikeus. Riskien arviointiin tulisi käyttää riittävän laajaa tilastodataa, joka itsessään vie organisaatiolta resursseja, eikä esimerkiksi pienemmillä yrityksillä ole välttämättä varaa panostaa siihen.

2.3 Palautumissuunnitelma

Jokaisella organisaatiolla tulisi olla voimassa oleva palautumissuunnitelma ennakoimattoman katastrofin varalta. Liiketoiminnan tai muun operatiivisen toiminnan jatkuminen on erittäin tärkeää, sillä palvelussa käyttökatko saattaa merkitä myyntitulojen, palvelun uskottavuuden ja markkinaosuuksien menetystä. Katastrofit voivat johtua esimerkiksi tahattomista laitteistovioista tai tahallisista palvelunestohyökkäyksistä (Denial of Service, DOS). Muun muassa näitä varten on organisaatioilla oltava käyttökelpoinen, testattava, skaalautuva ja ylläpidettävä palautumissuunnitelma. (Alhazmi & Malaiya 2012, s. 19)

On kuitenkin helppoa jättää riittävä palautumissuunnitelma laatimatta, sillä normaalissa tilanteessa sitä ei tarvita. Suunnitelman laatiminen ja varsinkin testaaminen saattaa olla kallista ja aikaa vievää, eikä sille ole varattu tarvittavia resursseja. Palautumissuunnitelmalle ei ole myöskään helposti laskettavissa takaisinmaksuaikaa tai sijoitetun pääoman tuottoastetta. Sama ongelma koskee myös muita järjestelmien kahdennuksia tai poikkeustilanteisiin varautumisia, sillä ne eivät tuota suoranaisesti mitään, eikä niitä välttämättä tulla missään vaiheessa tarvitsemaan. Kunnollinen palautumissuunnitelma voi kuitenkin osoittautua katastrofitilanteessa korvaavamattomaksi työvälineeksi.

Palautumissuunnitelman voisi ajatella olevan myös työkalu organisaation tietohallinnolle, jolla se perustelee liiketoiminnalle välttämättömiksi katsovansa investoinnit. Suunnitelmassa siis konkretisoituu se, mihin järjestelmä nykyisellään kykenee poikkeustilanteen sattuessa. Tällöin organisaation liiketoimintayksikkö voi peilata näitä tuloksia omiin tarpeisiinsa ja tekemiinsä riskiarviointeihinsa.

Perinteisesti palautumissuunnitelmat ovat perustuneet kahden identtisen konesalin (ammattiterminologiassa saitti, engl. site) ylläpitämiseen, joista toinen on ensisijainen ja toinen varalla. Fyysisesti näiden konesalien tulisi sijaita myös jonkin matkan päässä toisistaan, jotta esimerkiksi mahdolliset luonnonkatastrofit eivät vaikuta molempiin samanaikaisesti. Varsinkin pienille organisaatioille tämä lisäisi toiminnan kustannuksia niin merkittävästi, että harvinaisiin poikkeustilanteisiin ei haluta tai ole edes mahdollista varautua. Tähän ongelmaan on kuitenkin nykyisin olemassa kustannustehokkaampia ratkaisuja, sillä varakonesalin virkaa voi toimittaa julkinen pilvipalvelu, jossa kustannukset muodostuvat käytön mukaan. (Alhazmi & Malaiya 2012, s. 19) Julkisen pilven käyttöä saattaa kuitenkin rajoittaa organisaation omat tietoturvallisuusperiaatteet tai yrityksen toimintaa koskeva voimassa oleva lainsäädäntö. Julkisten pilvipalveluiden hyödyntäminen osana palautumissuunnitelmaa saattaa kuitenkin monelle pienelle orga-

nisaatiolle olla täysin riittävä vaihtoehto. Poikkeustilanteen varalta tulee palautumissuunnitelma olla kuitenkin riittävän kattavasti testattu, jotta organisaatio voisi jatkaa normaalia toimintaa mahdollisimman nopeasti.

Palautumissuunnitelmaan vaikuttaa olennaisesti myös käytettävissä olevat palvelinten varmistusjärjestelmät, sillä näiden kyky palauttaa ja varmistaa dataa heijastuu suoraan mahdollisiin varakonesaliratkaisuihin. Varmistusjärjestelmään liittyy kaksi helposti mitattavissa olevaa suuretta: palautumispiste ja toipumiskyky, joita käsitellään hieman tarkemmin seuraavassa luvussa.

2.4 Palautumispiste ja toipumiskyky

Varmuuskopioita ei oteta palvelimista pelkästään käyttäjien vahingossa poistamia yksittäisiä tiedostoja varten, vaikkakin tämä on ainakin toivottavasti varmistusjärjestelmän yleisin käyttötapaus. Katastrofipalautukseen voidaan joutua konesalin täydellisen tuhoutumisen tapauksessa, esimerkiksi tulipalon, vesivahingon, maanjäristyksen tai muun luonnonilmiön seurauksena. Myös kriittisen järjestelmään kohdistuva laiterikko saattaa vaatia katastrofipalautuksen suorittamisen toiminnan jatkamiseksi. Tällöin järjestelmä tulisi olla palautettavissa käyttöön palvelutasosopimuksen (SLA) mukaisesti. Palautettava tila ei myöskään saa olla liian vanha. Näitä vaatimuksia kuvastavat termit palautumispiste (Recovery Point Objective, RPO) sekä toipumiskyky (Recovery Time Objective, RTO), jotka ovat osa organisaation palautumissuunnitelmaa.

Palautumispiste voidaan määritellä kahden peräkkäisen varmuuskopion välisenä aikana. Tämä määrittää myös katastrofin seurauksena menetetyn datan maksimimäärän. Päivittäin otettujen varmuuskopioiden tapauksessa tämä on siis 24 tuntia. Toisaalta esimerkiksi kahdennettu replikoiva konesaliratkaisu mahdollistaa teoriassa sen, että palautumispiste on nolla. (Alhazmi & Malaiya 2012, s. 20). Käytännössä tähän ei kuitenkaan aina päästä, sillä vaikka käytössä olisi reaaliaikaisesti replikoiva levyjärjestelmä kahdessa fyysisessä eri sijainnissa, niin varalla olevan levyjärjestelmän käyttöönotto ei välttämättä ole mahdollista ilman tuotantokatkoa. Tämän lisäksi data on voinut korruptoitua tai pyyhkiytyä myös varalla olevasta levyjärjestelmästä replikoinnin seurauksena. On siis muistettava, että datan replikointi ei korvaa missään nimessä täysin varmuuskopioiden ottamista, mutta mahdollistaa joissain tapauksissa nopean toipumisen.

Toipumiskyky kuvaa liiketoiminnan tai muun operatiivisen toiminnan palauttamiseen kuluvaa aikaa, eli aikaa joka kuluu katastrofin ilmenemisestä toiminnan palauttamiseen normaaliksi. Laajojen järjestelmien osalta katastrofipalautuksessa saattaa kestää useita päiviä tai alle minuutti, riippuen käytetystä varmistusmediasta ja käytettävissä olevasta tietoliikenneinfrastruktuurista. (Alhazmi & Malaiya 2012, s. 19-20)

2.5 Arkistointi ja varmuuskopiointi

Arkistointi ja varmuuskopiointi ovat helposti sekoitettavissa toisiinsa, mutta tarkoittavat todellisuudessa eri asioita. Termien sekoittamisen lisäksi saatetaan myös itse menetelmiä käyttää väärin. Datasta saatetaan ottaa varmuuskopioita, vaikka tarve olisi oikeasti informaation arkistoinnille. Tässä työssä on käytetty tarkoituksella varmuuskopioinnin yhteydessä termiä *data* korostamaan menetelmien eroja. Varmuuskopioinnissa saattaa olla kyseessä varsin rakenteettoman datan kopiointia mediasta toiseen. Sen sijaan arkistoinnissa data voidaan mieltää sisältävän selkeämmin tietynlaisen rakenteen, eli tässä tapauksessa olisi perusteltua käyttää termiä informaatio. Jatkossa tässä diplomityössä käytetäänkin tästä syystä termiä data, kun käsitellään varmuuskopiointia.

Paulsen määrittelee arkistoinnin datan pitkäaikaiseksi säilyttämiseksi. Arkistoinnin jälkeen data ei ole enää aktiivisesti käytössä ja on siten perusteltua siirtää se toiselle medialle. Data on kuitenkin edelleen tärkeää ja tulee säilyttää mahdollista myöhempää käyttöä varten. Arkistointi saattaa vaatia myös paikallisten säädösten noudattamista. Arkistointi voidaan tehdä esimerkiksi kiintolevyille, optiselle tai holografiselle medialle, puolijohdelevyille tai nauhoille. Varmuuskopiointi on sen sijaan määritelty kadonneen tai korruptoituneen datan palauttamista varten. Se on myös selkeästi dynaamisempaa, sillä varmistettava data muuttuu tiheämmin ja toisaalta luonteeltaan ennalta ehkäisevää. Varmuuskopioinnin avulla on myös mahdollista palata muuttuneen tiedoston aikaisempaan versioon. (Paulsen 2012, s. 332, 337)

Toimivan informaation arkistoinnin avulla helpottuu myös varmuuskopiointi, sillä aktiivisesti käytössä olevassa tietojärjestelmässä ei ole niin paljon varmistettavaa dataa. Samana pysyvää ja vuosia vanhaa dataa ei ole järkevää pitää varmistuksissa mukana, sillä tällä haaskataan helposti käytössä olevia resursseja. Suurien informaatiovarastojen hallitseminen manuaalisesti on kuitenkin erittäin työlästä, joten tarkoitusta varten on syytä harkita automatisoitua järjestelmää.

2.6 Ohjeita ja standardeja

Valtiovarainministeriön alainen Valtionhallinnon tieto- ja kyberturvallisuuden johtoryhmä (VAHTI) säätää, ohjaa ja antaa suosituksia julkishallinnon tietojärjestelmien, tietoverkkojen ja ICT-palveluiden ylläpitoon (Valtiovarainministeriö). Ulospäin kenties näkyvin osa ryhmän toimintaa ovat julkaistut VAHTI-ohjeet, jotka ovat yleensä suhteellisen yleisellä tasolla kirjoitettuja ohjeita valtionhallinnon käyttöön. Mikään ei kuitenkaan estä näiden ohjeiden soveltamista myös yksityisellä sektorilla ja soveltuvin osin myös erittäin pienissäkin organisaatioissa.

Esimerkiksi melko vanhassa VAHTI 1/2002 -ohjeessa mainitaan erikseen varmistusjärjestelmän sijainnista suhteessa kolmeen eri aikana otettuun varmuuskopioon. Ohjeen mukaisesti päivittäin, viikoittain ja kuukausittain otetut nauhavarmistukset tulevat olla

fyysisesti kolmessa eri sijainnissa ja näiden sisältö tulee olla dokumentoitu riittävällä tarkkuudella. (VAHTI 1/2002, s. 37) Ohjeesta näkyy selvästi, että varmistukset ovat perinteisesti toteutettu puhtaasti nauhoihin perustuvia varmistusjärjestelmiä käyttäen. Levypohjaisten varmistusjärjestelmien soveltaminen näihin ohjeisiin ei ole ihan niin suoraviivaista.

VAHTI 2/2013 -ohjeet korvaavat vuonna 2002 julkaistun aikaisemman ohjeen ja esimerkiksi tässä uudistuneessa ohjeessa ei mainita enää nauhoja varmistusmediana eikä päivittäisistä, viikoittaisista tai kuukausittaisista varmistuksista ole täten suoraa viittausta. Ohje on muutenkin muotoiltu entistä vapaammin tulkittavaksi, mutta esimerkiksi vaatimus kahden fyysisen eri tilan käyttämisestä on jätetty myös uuteen ohjeeseen. (VAHTI 2/2013, s. 71). Tämä antaa entistä vapaammat kädet varmistusten tekniselle toteuttamiselle, joka mahdollistaa uusien tekniikoiden käyttämisen palvelinten varmistamisessa ja siten mahdollisesti nopeammat datan varmistukset sekä palautukset.

Yhdysvalloissa NIST (National Institure of Standards and Technology) on kansallinen standardointiin ja tekniikkaan keskittynyt virasto, joka julkaisee nimellä FIPS (The Federal Information Processing Standards) sarjan erilaisia tietoturvallisuuteen kuuluvia standardeja ja ohjeita. FIPS 200 (Minimum Security Requirements for Federal Information and Information Systems) keskittyy määrittelemään todella yleisellä tasolla tietoturvallisuuteen liittyviä vaatimuksia, jossa varmuuskopiointi on vain mainittu osana valmiussuunnitelmaa (CP, Contingency Planning). Valmiussuunnitelmaan liittyy myös tässäkin diplomityössä sivuttu palautumissuunnitelma. (FIPS 200, s. ii, 3)

Tarkemmin varmuuskopioinnin ja palautumisen toteuttamiseen ottaa kantaa sen sijaan NIST:in julkaisema SP 800-34 (Contingency Planning Guide for Federal Information Systems), jossa korostetaan varmistusstrategian toteuttamista järjestelmän kriittisyyden mukaisesti. Kriittisyyteen vaikuttaa saavutettavuuden menettämisen vaikuttavuus, eli mikä on vaikutus käyttäjille, mikäli järjestelmä ei olekaan saavutettavissa. Tässä julkaisussa vähemmän kriittisille palveluille nauhavarmuuskopiot todetaan riittäväksi datan varmuuskopioinniksi ja tarvittaessa järjestelmä voidaan sellaisenaan siirtää toiselle saitille ensisijaisen saitin tullessa käyttökelvottomaksi. Sen sijaan erittäin kriittisille järjestelmille vaaditaan kahdennettuja järjestelmiä ja levyillä olevan datan replikointia. Järjestelmällä tulee olla myös valmiiksi täydellisen infrastruktuurin omaava ja nopeasti käyttöön otettava varakonesali. Varmistusmedioiksi hyväksytään levyt, nauhat ja optiset mediat. (SP 800-34, s. 20–22)

3. LAITESALI- JA VARMISTUSJÄRJESTELMÄ-TEKNIIKKA

Tässä luvussa esitellään keskeisimmät laitesaleissa käytössä olevat ja varmistusjärjestelmien käyttämät teknologiat, käytännöt ja termit. Puhtaasti teknisistä ratkaisuista syvennytään levyjärjestelmiin ja niissä käytettävään SAN-verkkoihin (Storage Area Network), virtuaalipalvelimiin sekä deduplikointiin. Näistä erityisesti viimeiseksi mainittu edustaa varsin tuoretta teknologiaa, jolla pyritään minimoimaan varmistuksissa käytetty levytila sekä helpottamaan varmuuskopioiden siirtämistä hitaan WAN-verkon (Wide Area Network) yli.

Varmuuskopioinnissa on käytössä myös lukuisia joukko vakiintuneita käytäntöjä, joilla ei tosin ole välttämättä vakiintunutta suomenkielistä termiä. Tässä luvussa tarkastellaan myös erilaisia varmistusten tyyppejä (inkrementaalinen, differentiaalinen ja täysi), palautumispistettä ja toipumiskykyä, nauhakiertoja, varmistusten aikataulutusta sekä varmistusten järjestämistä (engl. staging).

3.1 Varmistusmenetelmät

Varmistusmenetelmillä tarkoitetaan tässä työssä kolmea eri tapaa ottaa varmuuskopio järjestelmän tiedostoista. Nämä kolme tapaa ovat täysi (engl. full), inkrementaalinen (engl. incremental) ja differentiaalinen (engl. differential) varmistus. Näitä varmistusmenetelmiä voidaan käyttää myös rinnakkain, mutta käytettävät menetelmät tulee valita siten, että se tukee organisaation määrittämää palvelutasosopimusta (SLA, Service Level Agreement) ja tämän pohjalta luotua varmistusstrategiaa.

Täysi varmistus kopioi aina kaikki valitut hakemistot ja tiedostot riippumatta siitä, että onko niihin edellisen varmuuskopioinnin jälkeen tullut muutoksia (Symantec 2012, s. 528). Tämä ei kuitenkaan tarkoita, että koko palvelimen sisällöstä otettaisiin varmuuskopio, vaan valittuna voi olla rajallinen joukko tiedostoja ja hakemistoja. Täysi varmistus vie eniten tilaa varmistusmedioilta ja on täten myös usein hitain suorittaa, varsinkin hitaan verkon yli. Sen sijaan esimerkiksi katastrofipalautuksen tekeminen nauhapohjaisesta järjestelmästä yksittäiselle palvelimelle on nopeinta suorittaa täydestä varmuuskopiosta, sillä palautettava data löytyy tällöin yhdeltä medialta.

Differentiaaliseen varmistukseen sisältyy kaikki ne tiedostot, jotka ovat muuttuneet edellisen täyden varmistuksen jälkeen. Inkrementaalinen varmistus sen sijaan kopioi vain edellisen täyden tai inkrementaalisen varmistuksen jälkeen muuttuneet tiedostot.

Muuttuneiden tai uusien tiedostojen selvittämiseen voi varmistusjärjestelmä käyttää esimerkiksi tiedoston muutoksen ilmaisevaa metatietoa, varmistuksen yhteydessä asetettavaa arkistointibittiä (engl. archive bit), käyttöjärjestelmän tiedostojen muutospäiväkirjaa (engl. change journal) tai varmistusjärjestelmän omaa tiedostoluetteloa. (Symantec 2012, s. 529–530) Differentiaalinen ja inkrementaalinen varmistusmenetelmä vähentävät varsinkin staattisissa ympäristöissä merkittävästi varmistettavan datan määrää ja lyhentävät varmistukseen kuluvaa aikaa, mutta lisäävät samalla hieman kompleksisuutta.

Inkrementaalinen varmistusmenetelmä käyttää vähiten tilaa varmistusmedioilta ja on yleensä myös nopein suorittaa, mutta kokonaisen palvelimen tai joissain tapauksissa jopa useita tiedostoja sisältävän hakemiston palauttaminen on tämän avulla hitaampaa, sillä eri päivinä muuttuneet tiedostot tulee palauttaa eri päivän varmuuskopioista, jotka saattavat sijaita jopa eri varmistusmedioilla. Differentiaalista varmistusmenetelmää käytettäessä vastaava palautus onnistuu aina käyttämällä korkeintaan kahta eri mediaa. (Symantec 2012, s. 530) Yksittäisen tiedoston palauttaminen onnistuu aina yhdeltä varmistusmedialta ja tämä onkin monesti yleisin palauttamisen tarve, joten tämä osaltaan puoltaa inkrementaalisen varmistusmenetelmän käyttöä.

Yleinen käytäntö on suorittaa täysi varmistus kriittisistä järjestelmistä kerran viikossa ja tämän lisäksi päivittäin inkrementaalinen tai differentiaalinen varmistus. Tällöin esimerkiksi kokonaisen palvelimen palauttamiseksi viimeisimpään tilaan saadaan suoritettua käyttäen korkeintaan seitsemää varmistusmediaa.

3.2 Nauhakierrot

Nauhakierrolla (engl. media rotation) tarkoitetaan tässä datan varmistamiseen tarkoitettujen nauhojen järjestelmällistä kiertoa, jolla voidaan hallita nauhojen kulumista. Varmistus-nauhat joutuvat fyysiseen kosketukseen kirjoitus- ja lukupään kanssa, joten on luonnollista, että nauhojen elinikä on rajallinen.

Yksinkertaisin nauhakierron tapa on käyttää yhtä ja samaa nauhaa päivittäin täyden varmuuskopion ottamiseen. Menetelmä on yksinkertainen hallittavaksi eikä tarvitse kuin yhden nauhan käyttöön, mutta saman nauhan jatkuva käyttäminen kuluttaa nauhaa ja tämä tulee siten aikanaan käyttökelvottomaksi. Tällöin varmistettu data sijaitsee myös vain yhdellä ainoalla nauhalla ja palautettavissa on vain viimeisin varmuuskopio järjestelmästä. (Symantec 2012, s. 398–399) Alalla vakiintuneena terminä tästä nauhakierron menetelmästä on poika (engl. son).

Inkrementaalista tai diffenrentiaalista varmistusmenetelmää, yhdessä viikoittaisen täyden varmistuksen kanssa, käytettäessä voidaan käyttää kuuteen nauhaan perustuvaa nauhakiertoa, jossa maanantaista torstaihin suoritettaville inkrementiaalisille tai differentiaalisille varmistuksille käytetään kullekin omaa nauhaansa ja perjantaina suoritet-

tavalle täydelle varmistukselle vuorotellen kahta nauhaa. Tällöin nauhojen kuluminen on tasaisempaa ja data on palautettavissa pidemmältä aikaväliltä, sillä nauhoille voidaan määrittää tässä tapauksessa kahden viikon päällekirjoitussuoja. (Symantec 2012, s. 399–400) Tämäkään nauhakiertomenetelmä ei kuitenkaan välttämättä ole organisaation palvelutason kannalta riittävä, sillä esimerkiksi vahingossa poistettua dataa ei välttämättä huomata riittävän ajoissa. Tämä nauhakiertomenetelmä tunnetaan alalla termillä isä-poika (engl. father-son).

Edellisiä kattavammassa ja erittäin yleisesti käytetyssä nauhakiertomenetelmässä käytetään edellisen tapauksen lisäksi kuukausittaisia nauhoja, jotka varmuuskopioinnin jälkeen tuodaan pois konesalista. Tällaista nauhaa kutsutaan off-site-nauhaksi. Tällöin siis esimerkiksi neljälle nauha-medialle otetaan päivittäiset, kolmelle viikoittaiset ja 12 nauhalle kuukausittaiset varmistukset, jolloin tällä nauhakierrolla saadaan jo melko hyvä kattavuus ja tiedon palautettavuus viimeisen vuoden ajalta. (Symantec 2012, s. 400–401) Kuukausittaiset nauhat voidaan myös jättää kokonaan kierrättämättä, jolloin järjestelmässä on jo hieman datan arkistoinnin piirteitä. Edellisistä nauhakierroista on helppo johtaa tälle menetelmällä myös oma termi, isoisä-isä-poika (engl. grandfather-father-son).

Edellä kuvatut nauhakierrot ovat vain muutamia esimerkkinä mainittuja, joista voi nähdä, miten erilaisia datan varmistukseen liittyviä tarpeita voi eri organisaatioilla olla. Esimerkiksi terveys- ja finanssialalla saattaa jo laki määrittää sellaisia tarpeita, joita ei pysty täyttämään edellä kuvatuilla varmuuskopioinnin menetelmillä. Näihin tarpeisiin linkittyy myös monesti vahvat järjestelmien kahdennukset.

Nauhakierron hallinta on onneksi myös helppo automatisoida ja monet nauhakirjastot pystyvät huolehtimaan tästä täysin itsenäisesti. Tällöin vain kuukausittaisten nauhojen tuonti pois konesalista vaatii järjestelmän ylläpitäjältä nauhakirjaston luona käynnin.

3.3 Levyjärjestelmät ja SAN-verkot

Levyjärjestelmät ja SAN-verkot ovat merkittävässä asemassa nykyajan laitesaliympäristössä. Levyjärjestelmä ja SAN-verkot liittyvät usein myös kiinteästi palvelinten varmuuskopiointiin. Erityisesti virtuaalipalvelinten kanssa on luonnollista käyttää erillistä levyjärjestelmää isäntäkoneiden paikallisen levytilan sijasta. Tällä saadaan aikaiseksi helpommin ylläpidettävä järjestelmä. Levyjärjestelmät puolestaan ovat yleensä kytketty virtuaalipalvelinympäristöön käyttäen tarkoitusta varten suunniteltuja SAN-verkkoteknologioita. SAN-verkoilla ei viitata yksittäiseen protokollaan, vaan kyseisen käsitteen alle mahtuu suuri joukko erilaisia protokollia ja kytkentätapoja.

3.3.1 Levyjärjestelmät

Levyjärjestelmä terminä ei suomen kielessä ole täysin vakiintunut, mutta tässä diplomityössä sillä tarkoitetaan erillistä kyseiseen tarkoitukseen dedikoitua tallennusjärjestelmää, jonka avulla levykapasiteettia voidaan joustavasti provisioida sitä tarvitseville järjestelmille, jotka konesaliympäristössä ovat tyypillisesti palvelimia. Levyjärjestelmä koostuu siis joukosta kiintolevyjä tai puolijohdetekniikkaan perustuvia SSD-levyjä (Solid State Drive), joita samassa levyjärjestelmässä voi olla myös yhdessä. Levyjä ohjaa vähintään kaksi toisistaan riippumatonta levyohjainta, joilla saavutetaan järjestelmän korkeat saatavuusvaatimukset. Levyjärjestelmä ja sen resursseja käyttävä palvelin kytketään toisiinsa nopealla SAN-verkolla.

Levyjärjestelmän avulla palvelimille pystytään antamaan aina kulloinkin tarvittava määrä levykapasiteettia. Tällöin tarpeen muuttuessa ei itse palvelimelle tarvitse hankkia ja asentaa uusia levyjä, vaan tälle vain osoitetaan olemassa olevasta levyjärjestelmästä tarvittava määrä levytilaa. Koska levytilaa voidaan provisioida tarkasti palvelimen tarvitsema määrä, saavutetaan organisaatiotasolla myös korkeampi levykapasiteetin konsolidointiaste. Aina ei myöskään ole kannattavaa tai edes mahdollista varustaa palvelimia tarvittavalla määrällä levyä. Näin on varsinkin niin sanottujen korttipalvelinten (engl. blade server) osalta tilanne, joihin ei tyypillisesti saa kuin kaksi 2,5 tuuman kiintolevyä asennettua niiden pienen kokonsa vuoksi.

Tässä työssä pyritään tekemään selkeä ero levyjärjestelmien ja NAS-verkkotallennusjärjestelmien välillä (Network Attached Storage), joissa verkon yli tarjotaan tiedostotasolla kapasiteettia verkon eri päätelaitteille. Näissä datan käsittelyn abstraktiotasot poikkeavat toisistaan.

Dapeng et al. on vertaillut NAS- ja SAN-tallennusratkaisujen suorituskykyjä keskenään, mutta samalla artikkelissaan määritellyt näiden ratkaisujen tekniset erot melko selkeästi. NAS-tallennusratkaisuilla tarkoitetaan esimerkiksi NFS-protokollaa (Network File System) käyttävää ratkaisua, jossa asiakasjärjestelmät pystyvät käsittelemään dataa verkkotallennuslaitteelta vain tiedostotasolla. Sen sijaan SAN-pohjaista järjestelmää käyttävät järjestelmät, joissa käytettynä protokollana on esimerkiksi iSCSI, käsittelevät dataa blokkitasolla ja siten käyttävät verkon yli jaettua resurssia kuin paikallista levyä. (Dapeng et al. 2009, s. 30) Tässä työssä levyjärjestelmällä tarkoitetaan edellä mainituista jälkimmäistä ratkaisua, eli SAN-pohjaisia järjestelmiä.

Terminä NAS on siis hieman harhaanjohtava, sillä SAN-järjestelmät ovat yhtä lailla verkolla kytkettyjä tallennusratkaisuja. NAS-laitteen levyille ei voi asentaa asiakkaan käyttöjärjestelmää, joten tämä rajoittaa sen käyttökelpoisuutta palvelinympäristössä. Sen sijaan esimerkiksi varmistusjärjestelmän mediana NAS-järjestelmää on mahdollista käyttää.

Erittäin korkean saavutettavuuden tarpeisiin levyjärjestelmän data replikoidaan toisessa fyysisessä sijainnissa olevan konesalin levyjärjestelmään, jolla voidaan varautua esimerkiksi luonnonkatastrofin aiheuttamiin tuhoihin.

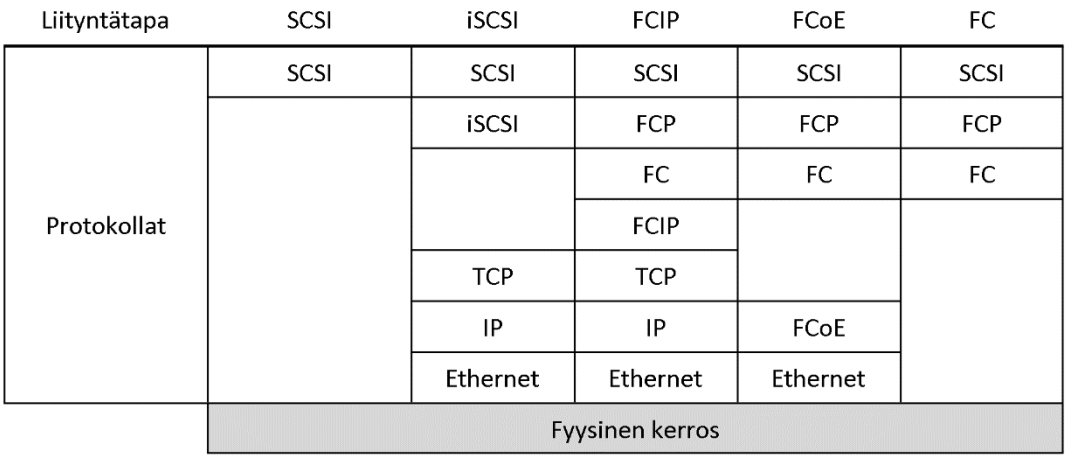
3.3.2 SAN-verkot

SAN-verkot on suunniteltu yhteensopiviksi eri laitevalmistajien välillä ja tätä yhteensopivuutta pyrkii edistämään SNIA (The Storage Networking Industry Association), joka on eri laitevalmistajien edustajista koottu voittoa tuottamaton yhdistys. SNIA on keskittynyt erityisesti juuri tallennusratkaisujen teknologioiden, standardien ja koulutuksen kehittämiseen ja sillä on jäseninä noin 400 alan yritystä. (SNIA)

Tallennusjärjestelmien liityntä palvelimiin voidaan toteuttaa monella tavalla. Yksinkertaisin tapa on kytkeä levy, kokonainen levyjärjestelmä tai nauhuri suoraan palvelimeen. Liityntätapana voi tällöin olla esimerkiksi INCITS:in (InterNational Committee on Information Technology Standards) kehittämä ja ANSI:n (American National Standards Institute) standardoima rinnakkainen SCSI-liityntä (Small Computer System Interface) tai nykyaikaisempi sarjamuotoinen tiedonsiirtotapa SAS (Serial Attached SCSI), joka yksinkertaistaa kaapelointia. SCSI ei kuitenkaan viittaa pelkästään fyysiseen tiedonsiirtomuotoon, vaan se on myös looginen standardi sisältäen myös joukon käskyjä (CCS, Common Command Set), joita esimerkiksi myöhemmät sarjamuotoiseen liikennöintiin perustuvat optiset kuituverkot hyödyntävät. (INCITS; IBM 2012, s. 20–22, s. 32–34)

Mikäli yksittäisen levyjärjestelmän tai nauhurin resurssit halutaan käytettäväksi useamman palvelimen kesken, kannattaa nämä kytkeä samaan SAN-verkkoon. Tämä mahdollistaa niiden monipuolisemman hallinnan ja yksinkertaisemman johdotuksen, sillä esimerkiksi levyjärjestelmää ei tarvitse kytkeä suoraan yksittäisiin palvelimiin. SAN-verkkojen eri toteutukset muistuttavatkin hyvin paljon OSI-mallia tai nykyisen internetin toimintaa kuvaavaa TCP/IP-viitemallia (Transmission Control Protocol/Internet Protocol).

Protokollapinossa ylimpänä on SAN-verkoissa aina SCSI ja tämän alapuolella olevat protokollat vaihtelevat käytetyn siirtotien ja teknologiavalintojen myötä. Kuvassa 3.1 vertaillaan eri tallennusjärjestelmien ja SAN-verkkojen kytkentätapojen käyttämiä protokollia ja niiden järjestystä protokollapinossa.



Kuva 3.1. *SAN-verkkojen kytkentätätapojen käyttämät protokollat (IBM 2012, s. 23).*

Kuten kuvasta voidaan havaita, vain SCSI, Ethernet ja FC (Fibre Channel) toimivat suoraan fyysiseen kerroksen päällä, joka voi olla myös optisen kuidun lisäksi kuparikaapeli. Kuvassa 3.1 esitetty SCSI-liityntätapa viittaa tässä tapauksessa vanhaan rinnakkaiseen tiedonsiirtoon perustuvaan liitäntään, jossa laitteet tallennusjärjestelmä on suoraan kytketty palvelimeen (engl. server-attached storage).

Kolme seuraavaa liityntätapaa (iSCSI, FCIP ja FCoE) mahdollistavat esimerkiksi käytössä olevan TCP/IP- tai Ethernet-verkon käyttämisen siirtoverkkona, jolla pystytään toteuttamaan verkko kustannustehokkaammin. (IBM 2012, s. 23–25) Useampi päällekkäinen protokolla vähentää kuitenkin hyötykuorman suhdetta siirrettävän datan määrään, ja siten heikentää verkon suorituskykyä. Toisaalta vanhaa SCSI-liityntää käytettäessä suurin tiedonsiirtonopeus on vain 160 megatavua sekunnissa (Ultra 160/m) ja kaapelin pituuskin varsin rajallinen, kun vastaavasti Ethernet-verkkojen nopeudet mitataan nykyään gigabiteissä (IBM 2012, s. 22; IEEE 802.3).

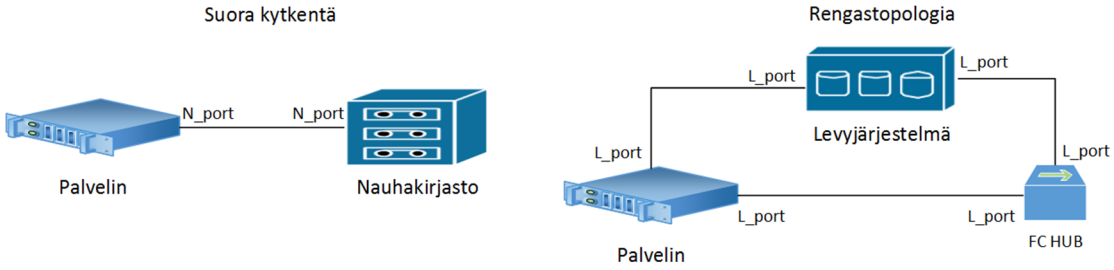
Kuvassa 3.1 esitetty iSCSI (Internet Small Computer System Interface) kuljettaa SCSI-pyynnöt ja datan TCP/IP-verkon päällä, jolloin voidaan samoilla verkkolaitteilla siirtää sekä tallennusjärjestelmien että palvelimilla toimivien sovellusten tarvitsema data monella erilaisella fyysisellä siirtoverkolla.

FCP-protokolla on rajapintaprotokolla FC:n ja SCSI:n välissä. FCIP (Fibre Channel over IP) käyttää myös TCP/IP-verkkoa datan siirrossa, mutta jättää protokollapinon ylemmän puolen vastaamaan FC-liityntätavan käyttämiä protokollia. Tämä mahdollistaa olemassa olevien FC-pohjaisten SAN-verkkojen tunneloinnin tavallisessa TCP/IP-verkossa. (IBM 2012, s. 24)

Kuitua käyttävissä SAN-verkoissa voidaan käyttää hyvin monenlaisia fyysisiä topologioita. Yksinkertaisimmillaan kaksi kuituliitäntäistä laitetta voidaan kytkeä suoraan toi-

siinsa (engl. point to point). Tällöin linkin muodostaminen on yksinkertaista ja kaista taattua. FC-verkon nopeudet on standardisoitu neljään eri nopeusluokkaan, jotka ovat 2, 4, 8 ja 16 gigabittiä sekunnissa. Tiedon siirto on kaksisuuntaista, eli kummatkin osapuolet voivat lähettää ja vastaanottaa samanaikaisesti. (IBM 2012, s. 86)

Korkeintaan 126 noodia voivat luoda keskenään rengastopologian (FC-AL, Fibre Channel Arbitraded Loop), jossa data ja ohjauskomennot kiertävät yhteyden muodostamisen jälkeen renkaassa aina yhteen suuntaan. Käytössä ovat 2 ja 4 gigabitin nopeus-luokat, mutta luonnollisesti viive kasvaa sitä suuremmaksi, mitä enemmän laitteita renkaaseen on liitettynä. Tämä topologia mielletään myös vanhahtavaksi eikä ole laajalti enää käytössä. (IBM 2012, s. 87) Kuvassa 3.2 on esitetty esimerkkitapaukset suoraan kytketyistä kuituverkon laitteista sekä rengastopologian muodostavasta verkosta.

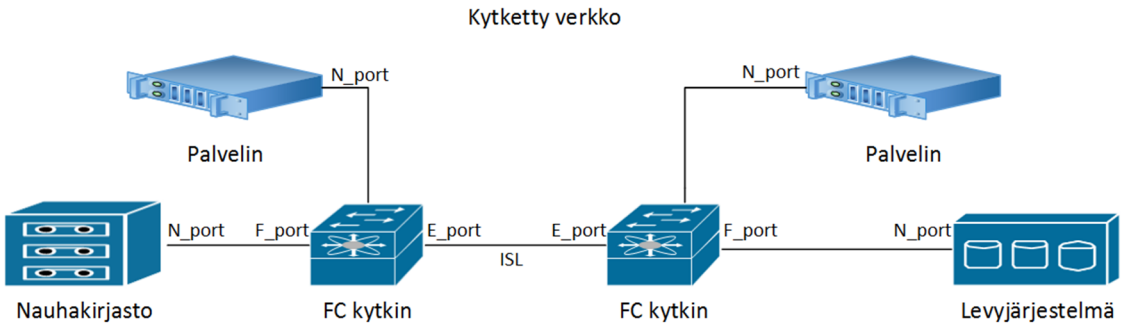


Kuva 3.2. *Suora kytkentä ja rengastopologia SAN-verkoissa (IBM 2012, s. 86 – 87).*

Monipuolisin topologia saadaan muodostettua käyttämällä kytkettyä verkkoa (FC-SW, Fibre Channel Switched Fabric). Tällöin verkossa on osallisena vähintään yksi aktiivilaite, joka kytkee eri noodeista tulevia paketteja. Kullakin noodilla on linkin täysi kapasiteetti käytettävissään ja koko verkon kapasiteetti on kaksisuuntaista tiedonsiirtoa tukevissa laitteissa yksittäisten linkkien nopeuksien summa. Useampi kytkin voidaan liittää kytkinten välisillä linkeillä (ISL, Inter-Switch Link) sarjaan (kuva 3.3) tai niistä voidaan luoda oma rengas, jolla saavutetaan vikasietoisuutta.

Mikäli kaikki kytkimet ovat kytketty suoraan toisiinsa, on kyseessä mesh-topologia (engl. mesh topology). Kytkinten lisäksi FC-SW-verkkojen yhteydessä kuulee mainittavan ohjaajista (engl. director), jotka ovat suurempia kytkimiä ja omaava tavanomaisiin kytkimiin verrattuna enemmän redundanssia, mutta joiden rooli verkossa on kuitenkin hyvin lähellä kytkimiä. (IBM 2012, s. 88 – 92; 149)

Näitä edellä mainittuja topologioita yhdistelemällä saadaan myös monimutkaisempia topologioita aikaiseksi, mutta näiden tarkastelu rajataan tämän diplomityön ulkopuolelle.



Kuva 3.3. *Kahden kytkimen muodostama kytketty verkko (IBM 2012, s. 88 – 92).*

Kuituverkkojen yhteydessä on kytkinten, palvelinten ja erilaisten tallennuslaitteiden käyttämille porteille annettu niiden merkityksensä mukaisesti nimet. Seuraavassa on listattu muutamia tavallisimpia portteja ja kerrottu niiden merkitys.

- **N_port**: palvelimessa tai tallennuslaitteessa oleva portti
- **F_port**: kytkimessä oleva portti, jonka suorana vastineena on palvelin tai jokin tallennuslaite (N_port)
- **E_port**: kytkimessä oleva portti, jonka avulla voidaan muodostaa suurempi verkko kytkemällä kaksi kytkintä yhteen
- **G_port**: geneerinen portti, joka voi toimia sekä linkin muodostamisen jälkeen kuten F_port että E_port
- **L_port**: rengastopologiaan soveltuva palvelimen tai kytkimen portti (IBM 2012, s. 93–94).

Kuituverkoissa eri laitteet tunnistetaan uniikilla 64 bittiä pitkällä tunnisteella, maailmanlaajuisella nimellä (WWN, World Wide Name). Tässä nimikäytännössä IEEE hallinnoi WWN-nimestä valmistajaa yksilöivää osuutta ja valmistajien vastuulle jää, ettei heidän määrittämänsä osuus nimestä aiheuta duplikaatteja nimiä. Lisäksi WWN-nimet ovat jaettu kahteen alaryhmään, maailmanlaajuiseen noodinimeen (WWNN, World Wide Node Name) sekä porttinumeroon (WWPN, World Wide Port Number). Nämä eroavat toisistaan siten, että noodinimellä viitataan yhteen levyjärjestelmään, palvelimen HBA-sovittimeen (Host Bus Adapter) tai kytkimeen ja porttinumerolla viitataan yksittäiseen levyjärjestelmän, HBA-sovittimen tai kytkimen porttiin. (IBM 2012, s. 97–99)

WWN-nimien lisäksi kytketyssä verkossa käytetään kuitukytkimien ylläpitämiä 24-bittisiä porttiosoitteita (engl. port address), joiden avulla datan reitittäminen saadaan nopeammaksi sekä tällä voidaan vähentää manuaalista ylläpitoa. Porttiosoitteiden jakamisesta vastaa kytkimessä toimiva nimipalvelin, eivätkä nämä osoitteet ole globaalisti uniikkeja. 24-bittisen osoitteen bitit 23–16 kertovat toimialueen (engl. domain), johon kuitukytkimeen kytketty laite kuuluu. Toimialueella viitataan kulloinkin kyseessä olevaan kytkimeen ja kaikilla samaan kytkimeen liitetyillä laitteilla nämä bitit ovat samat.

Bitit 15–8 kertovat, että mihin kytkimen porttiin laite on kytketty. Loput bitit 7–0 (arbitraded loop physical address, AL_PA) yksilöivät portin rengastopologiassa. Mikäli viimeinen tavu on heksadesimaalina 00, niin kyseessä on suoraan kytkimeen liitetty laite, joka ei ole osallisena missään rengastopologiassa. Yksityisessä rengastopologiassa (engl. public loop), joka ei ole kytketty osaksi mitään muuta suurempaa verkkoa kytkimen avulla, ensimmäiset kaksi tavua ovat heksadesimaalina 0000. Tämä osoitteistus mahdollistaa täten rengastopologian liittämisen osaksi kytkettyä verkkoa. (IBM 2012, s. 101–103)

SAN-verkoilla on huomattavan paljon yhteisiä piirteitä IP-verkkojen kanssa. Esimerkkinä voidaan mainita protokollapino, joka muistuttaa monelta osin TCP/IP-mallin protokollapinoa. Myös WWN-nimien ja porttiosoitteiden välinen yhteys ja näiden muutos muistuttaa hyvin suurelta osin TCP/IP-verkkojen MAC- ja IP-osoitteita.

3.4 Virtuaalipalvelimet

Asiakas−palvelin-mallin sovellukset toimivat monesti keskitetysti yrityksen tai muun organisaation konesalissa. Useita palvelimia sisältävässä ympäristössä on hyvin monissa tapauksissa hyödyllistä käyttää virtualisoituja palvelimia, joiden avulla voidaan saavuttaa muun muassa merkittäviä kustannussäästöjä, sillä yhdellä fyysisellä palvelimella voi toimia useita virtuaalisia palvelimia, eikä täten jokaista sovellusta varten tarvita omaa fyysistä palvelinta.

Palvelinten virtualisointi ei ole kovinkaan uusi asia, sillä jo 1960-luvulla oli tätä tekniikkaa käytetty keskuskoneiden (engl. mainframe computer) resurssien jakamiseen. Tuolloin tarve virtualisoimiseen oli hieman erilainen kuin nykyään, sillä tuolloin laitteet olivat kalliita hankkia ja siten myös harvinaisia. Virtualisoimalla pystyttiin samalla fyysisellä koneella suorittamaan useampaa sovellusta. Tämän jälkeen kuitenkin virtualisointi jäi hieman taka-alalle, sillä uudet käyttöjärjestelmät mahdollistivat sovellusten moniajon ja samanaikaisesti laitteiden hinnat tippuivat merkittävästi, jolloin virtualisoinnista ei ollut enää niin suurta hyötyä. 2000-luvulla virtualisointi koki kuitenkin eräänlaisen renessanssin, sillä nykyään lukuisat suuret tietotekniikka-alan yritykset tarjoavat virtualisointiratkaisujaan markkinoille miljardien eurojen edestä. (Rosenblum & Garfinkel 2005)

Virtualisoinnissa käyttöjärjestelmä eristetään alla olevasta fyysisestä palvelinraudasta. Tällöin raudan ja käyttöjärjestelmän välissä toimii hypervisori (VMM, Virtual Machine Monitor), jonka ansiosta pystytään muun muassa virtuaalipalvelinten käyttämiä resursseja hallitsemaan tarkasti sekä tarvittaessa siirtämään virtuaalipalvelin kokonaan toiselle alustalle käyttöjärjestelmään huomaamatta mitään muutosta. Fyysisiä palvelimia voidaan tällöin käsitellä ikään kuin yhtenä suurena resurssipoolina. Hypervisori tarjoaa myös turvallisuutta, sillä yhden virtuaalipalvelimen kaatuminen tai vikaantuminen ei

vaikuta muihin samassa fyysisessä koneessa toimiviin virtuaalipalvelimiin. (Rosenblum & Garfinkel 2005)

VMwaren virtualisointialustalla voidaan yksittäinen virtuaalipalvelin siirtää toiselle fyysiselle palvelimelle, jota VMware nimittää ESXi-isäntäpalvelimeksi (ESXi host), manuaalisesti (vMotion) tai automaattisesti (DRS, Distributed Resource Scheduler) käytössä olevien resurssien muuttuessa. Virtualisointialusta antaa myös mahdollisuuden selviytyä yksittäisen ESXi-isäntäpalvelimen sammumisesta vikaantumistapauksessa siten, että virtuaalipalvelimet siirretään automaattisesti muille samassa korkean saavutettavuuden klusterissa (engl. high availability cluster) oleville toimiville ESXi-isäntäpalvelimille. (VMware)

Virtualisoinnin käyttäminen helpottaa huomattavasti palvelinympäristön ylläpitoa, sillä uusien sovellusten käyttöönotto ei useinkaan vaadi uuden fyysisen palvelimen asentamista ja käyntiä konesalilla. Resursseja voidaan myös joustavasti lisätä tai vähentää, mikäli palvelimen kuormitus muuttuu. Huollettavien laitteiden määrä myös pienenee ja virtuaalisia palvelimia saadaan enemmän mahtumaan samaan tilaan konesalissa. Virtuaalipalvelinten käytöllä voidaan myös saavuttaa säästöjä energian kulutuksessa. Itse palvelin ei ole ainoa energiaa kuluttava laite konesaliympäristössä, vaan merkittävän osan energiankulutuksesta aiheuttaa jäähdytyslaitteisto (Helin 2012, s. 7).

Tämän diplomityön varsinaiseen aiheeseen liittyen voidaan tässä vaiheessa todeta, että varmistusten ottaminen on määrätyissä tapauksissa helpompaa virtuaalipalvelinten osalta, sillä jokaiseen palvelimeen ei tarvitse välttämättä asentaa omaa agenttisovellusta varmistusten ottamista varten, vaan varmuuskopiointisovellus voidaan määrittää ottamaan varmistuksia kokonaisista virtuaalipalvelimista.

3.5 Varmistusten aikataulutus

Varmuuskopioinnin yhteydessä on vakiintunut tapa puhua varmistusten aikaikkunasta (engl. backup window), jolla tarkoitetaan säännöllisiä ajankohtia, jolloin varmistustyöt saisivat olla päällä. Tyypillisesti tämä aikaikkuna sijoittuu normaalien toimistoaikojen ulkopuolelle, jolloin esimerkiksi varmistustöiden aiheuttama LAN- ja SAN-verkkojen normaalia suurempi kuormittuminen ei häiritse organisaation tietojärjestelmien normaalia operatiivista toimintaa.

Mikäli varmistustyöt eivät mahdu annettuun aikaikkunaan ja pitkään kestäneet varmistustyöt häiritsevät muita järjestelmiä, tulee ylläpidon selvittää tähän johtavat syyt. Aina syynä ei ole yksinkertaisesti varmistettavan datamäärän kasvaminen, vaan syy voi olla verkkoyhteyksien toiminnassa. Myös varmistettavan datan tyyppi vaikuttaa varmistustöiden kestoon, kuten luvussa 5.3.1 esitetyssä tapauksessa käy selkeästi ilmi.

Mahdollista on myös karsia varmistettavaa dataa ja rationalisoida varmistustöitä, sillä esimerkiksi kaikista testipalvelimista ei välttämättä tarvitse olla yhtä tiheään otettuja varmuuskopioita kuin kriittisistä tietokantapalvelimista. Varmistusjärjestelmän tulisi kuitenkin aina täyttää organisaation tietoturvastrategiassa asetetut vaatimukset.

Jos kuitenkin varmistettavan datan määrä on kasvanut niin suureksi, että käytössä oleva järjestelmä ei sitä pysty annetussa aikaikkunassa käsittelemään, niin järjestelmää tulee päivittää vastaamaan muuttuneita tarpeita. Järjestelmän pullonkaula tulee täten selvittää ja mahdollisuuksien mukaan poistaa. Pullonkaulana voi toimia esimerkiksi LAN- tai SAN-verkko, jolloin verkon muuttaminen nopeammaksi on luonnollinen tapa ratkaista ongelma. Suurissa konesaliympäristöissä myös varmistusjärjestelmien kuormaa pystytään jakamaan useammalle palvelimelle tai nauhakirjastolle.

3.6 Varmistusten vaiheistaminen ja varmistusmediat

Konesalitekniikkaan liittyvä sanasto ei ole täysin vakiintunutta eikä kaikille termeille edes löydy suomenkielistä käännöstä. Näin on varsinkin tämän alaluvun otsikossa käytetyillä termillä, josta ei etsimällä löytynyt mitään viitteitä alan lähteistä. Varmistusten vaiheistamisella (engl. staging) tässä tapauksessa viitataan siis varmistettavan datan tallentamiseen lähdejärjestelmästä varmistusmedialle. Termi vaiheistus on kuitenkin kohtalaisen kuvaava, sillä monissa tapauksissa järjestelmän varmistuksen lopullinen sijainti ei ole sillä medialla, johon data on ensimmäisen kerran tallennettu, eli nopeampaa mediaa voidaan käyttää väliaikaisena datan säilytyspaikkana.

Nauhojen käyttöä varmistusmediana on puoltanut ja puoltaa edelleenkin niiden edullisuus levypohjaiseen tallentamiseen nähden. Niitä pidetään myös luotettavana pitkäaikaiseen datan säilömiseen. Nauhalla oleva data on myös helppo tuoda pois konesalista (engl. off-site backup), jolla pystytään varautumaan esimerkiksi kokonaisen konesalin tuhoutumiseen tulipalon tai vesivahingon varalta. Nauhan käyttäminen varmistusmediana on kuitenkin määrätyissä tilanteissa hitaampaa. (Tandberg 2013) Erityisesti yksittäisten tie-dostojen palauttaminen nauhalta saattaa olla hidasta. Tähän ongelmaan palataan tässä diplomityössä tarkemmin luvuissa 4.4.2 ja 6.3.

Nauhateknologiat kehittyvät lisäksi edelleen ja esimerkiksi kirjoitushetkellä viimeisin LTO-teknologiaan (Linear Tape Open) perustuva kuudes sukupolvi tarjoaa 2,5:1 suhteella pakattuna 6,25 teratavua kapasiteettia yhdellä nauhalla. LTO-yhtymällä on lisäksi suunnitelmissa jo neljä seuraavaa sukupolvea. (LTO)

Perinteisin tapa vaiheistaa varmuuskopiot on tallentaa data suoraan lähdejärjestelmästä nauhalle. Alalla vakiintuneena käytäntönä on tässä yhteydessä käyttää kirjain- ja numeroyhdistelmää D2T (Disk-to-Tape). Yksinkertaisin ratkaisu on kytkeä nauhuri suoraan varmistettavaan palvelimeen, mutta tällöin nauhurin tarjoamat resurssit eivät ole kovinkaan monipuolisesti hyödynnettävissä ja nauhojen käsittely vaatii manuaalista ylläpitoa

nauhakierron toteuttamiseksi. Laajemmassa konesaliympäristössä on tavallista kytkeä nauhuri tai useita nauhoja sisältävä nauhakirjasto erilliseen varmistuspalvelimeen joko suoraan tai SAN-verkon välityksellä. Tällöin varmistustöiden ylläpito on keskitetty ja täten helpompi toteuttaa.

Levypohjaisten tallennusratkaisujen kustannusten laskettua on tämän käyttäminen varmistusmediana tullut yhä kiinnostavammaksi ratkaisuksi. Myös esimerkiksi luvussa 3.7 esitelty deduplikointitekniikka on lisännyt levypohjaisten tallennusratkaisujen käyttökelpoisuutta ja kustannustehokkuutta. Alalla vakiintuneena lyhennyksenä käytetään tästä D2D (Disk-to-Disk). Tässäkin tapauksessa media voi olla suoraan varmistettavaan järjestelmään kytkettynä, mutta esimerkiksi varmistuspalvelimeen kytketty levypohjainen tallennusratkaisu on jälleen ylläpidon kannalta helpompi ratkaisu. D2D varmistuksista esimerkiksi yksittäisen tiedoston palauttaminen on nopeampaa kuin nauhalta, sillä nauhaa voidaan lukea vain yhdestä kohdasta kerrallaan, kun taas levyltä tiedosto voidaan välittömästi lukea mistä tahansa levyllä olevasta kohdasta (Tandberg 2013). Eli määrätyn yksittäisen tiedoston palauttamiseksi voidaan koko nauha joutua kelaamaan ja lukemaan alusta loppuun, kunnes oikea kohta on löytynyt. Tämän takia tiedostojen palauttamiseen kuluva aika on helpommin ennustettavissa käytettäessä D2D-tyyppistä datan varmistuksen vaiheistusta.

Käyttämällä näitä kahta edellä mainittua tapaa yhdessä, saavutetaan kummankin ratkaisun edut. Tämä vaiheistaminen tunnetaan lyhenteellä D2D2T (Disk-to-Disk-to-Tape). Levypohjaisella varmistuksella mahdollistetaan nopea varmuuskopion ottaminen, jolloin aikaikkuna on mahdollista saada pysymään lyhyenä. Tämän jälkeen levylle tallennettu data voidaan kaikessa rauhassa ja tuotantokäytössä olevia palvelimia häiritsemättä siirtää pitkäaikaisempaa tallennusta varten nauhalle. Levyllä olevista varmistuksista pystytään tällöin tarvittaessa hyvinkin nopeasti palauttamaan dataa ja nauhoilta dataa tarvitsee hakea vain tilanteessa, jossa palautettava data ei ole löydy levyltä. (Tandberg 2013) Levyn käyttäminen eräänlaisen väliaikaisena varastona on myös hyödyllistä tilanteessa, jossa varmistettava data on hitaan verkkoyhteyden takana. Tällöin verkkoyhteyden nopeus ei rajoita nauhurin nopeutta, vaan data voidaan kirjoittaa nauhalle vasta siinä vaiheessa, kun se on ensin kokonaisuudessaan haettu verkon yli, esimerkiksi organisaation sivukonttorissa olevasta työasemasta tai palvelimesta.

Levyihin ja nauhoihin perustuvan varmistuksen lisäksi on mahdollista tallentaa varmuuskopiot pilveen. Tätä kuvaava alalla vakiintunut lyhenne on D2C (Disk-to-Cloud) tai D2D2C (Disk-to-Disk-to-Cloud). Menetelmä soveltuu erityisesti palveluna ostettavaan varmuuskopiointiin, jossa asiakkaan ei tarvitse huolehtia itse varmistusjärjestelmän ylläpidosta, vaan ainoastaan varmistusten konfiguroinnista. Verkkoyhteyden nopeus rajoittaa kuitenkin usein varmuuskopioiden ottamista ja varmuuskopioiden määrittely poikkeaa monesti hieman LAN- tai SAN-verkon avulla toteutetuista varmuuskopiointiratkaisuista. Perinteisissä varmuuskopiointiratkaisuissa järjestelmistä otetaan säännöllisesti täydelliset varmuuskopiot, esimerkiksi kerran viikossa, ja päivittäin inkremen-

taaliset tai differentiaaliset varmistukset. Internet-yhteyden avulla ei kuitenkaan välttämättä ole mahdollista kerran viikossa ottaa järjestelmistä täydellistä varmuuskopioita, joten tällöin voidaan joutua ottamaan vain inkrementaalisia varmuuskopioita järjestelmästä. Myös tässä tapauksessa datan deduplikointi saattaa olla tarpeellinen tekniikka verkon yli siirrettävän datamäärän minimoimiseksi.

Mikäli järjestelmän datan määrä on niin suuri, että edes ensimmäistä täydellistä varmuuskopiota ei pysty kohtuullisessa ajassa siirtämään verkon yli pilveen, mutta päivittäiset muutokset järjestelmässä olevaan dataan ovat suhteellisen vähäisiä, voidaan ensimmäinen täydellinen varmuuskopio ottaa paikallisesti ja kuljettaa siirrettävällä massamedialla siirrettäväksi lopulliseen sijaintiinsa.

Pilvipohjaisen varmistusmedian käyttökelpoisuutta rajoittaa myös etenkin suurten datamäärien palauttamisen hitaus. Mahdollinen katastrofipalautus ei välttämättä ole mahdollista toteuttaa verkon yli liian pitkän toipumisajan vuoksi. Sen sijaan esimerkiksi yksittäisten inhimillisen erehdyksen seurauksena poistettujen tiedostojen palauttamiseen pilvipohjainen varmistusjärjestelmä sopii hyvin, erityisesti pienen organisaation käyttöön.

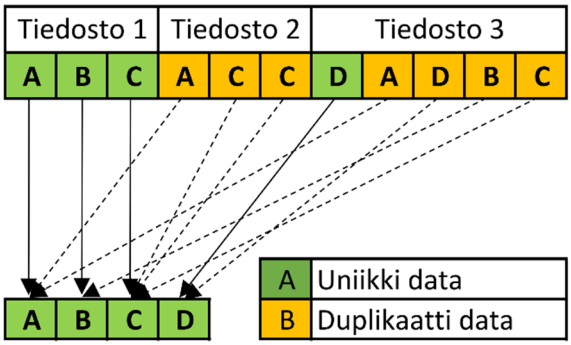
3.7 Deduplikointi

Kuten luvun 6 mittaustuloksista on huomattavissa, nauhaan perustuvassa varmuuskopioinnissa on merkittäviä heikkouksia, varsinkin datan palauttamisen nopeuteen liittyen. Toisaalta levypohjaisen varmuuskopioinnin käyttäminen perinteisillä menetelmillä ei ole monesti kustannusten takia mahdollista, vaikka hyödyt varsinkin datan palauttamisen osalta ovat ilmeiset. Tässä oletetaan siis, että varmuuskopiointiin ei riitä, että esimerkiksi kokonainen järjestelmä, yksittäinen tiedosto tai tietokanta pystytään palauttamaan edellistä varmuuskopiointia edeltävään tilanteeseen, vaan tarvittaessa jopa kuukausien takaiseen tilanteeseen. Tällöin yksi säilytettävä varmuuskopio ei riitä, vaan niitä pitää olla tietoturvapolitiikan vaatimusten määrittämältä ajalta ja halutulla tiheydellä.

Tähän ongelmaan merkittävän parannuksen tuo datan pakkaaminen käyttäen deduplikointitekniikkaa, joka on nykyään laajalti saatavissa eri valmistajien palvelinten varmistusratkaisuissa. Datan deduplikoinnilla voidaan tilanteesta riippuen päästä 95 prosentin levytilan säästöön varmuuskopioinnissa (Faritha Banu & Chandrasekar 2012, s. 364). Tällöin pakkaussuhde olisi siis jopa 1:20. Datan deduplikoinnilla säästetään valtavasti levytilaa, mutta siitä on hyötyä myös, mikäli tietoa varmistetaan hitaiden verkkoyhteyksien yli. Tällöin datan deduplikointi tehdään jo lähtöpisteessä eikä redundanttia dataa tarvitse siirtää verkon yli. Datan deduplikointi vaatii toisaalta runsaasti datan prosessointia, mikä ei nykyään yleisesti monesti ole kuitenkaan rajoittava tekijä.

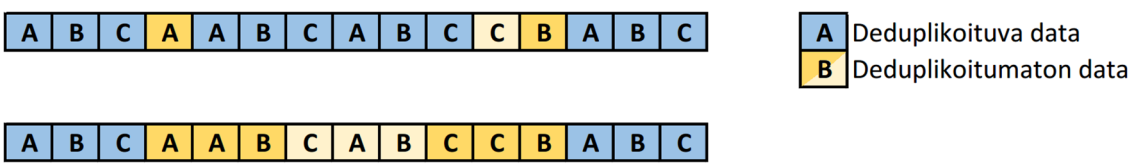
On kuitenkin huomattava, että datan deduplikointi ei ole sama kuin datan perinteinen pakkaaminen. Tässä datan pakkaaminen perustuu rajattuun joukkoon tiedostoja, jotka

pakataan kerralla. Datan pakkaamiseen on voitu käyttää esimerkiksi Huffman-koodausta, jossa datasta etsitään samanlaisia toistuvia osia, joihin viitataan jatkossa lyhemmällä koodilla (Huffman 1952). Tällöin tiedoston pakkaaminen on myös häviötöntä, eli data pystytään aina palauttamaan takaisin alkuperäiseen muotoonsa. Datan deduplikointia voidaan sen sijaan pitää prosessina, jossa tallennettava uusi data pilkotaan määrätyllä algoritmilla pienempii palasiin (engl. chunk), josta lasketaan tiiviste. Tätä tiivistettä verrataan tietokannassa oleviin aikaisemmin tallennetusta datasta muodostettuihin tiivisteisiin ja mikäli tiivisteiden tietokannasta löytyy vastaavuus, ei käsiteltävää palasta tarvitse tallentaa. Tällöin voidaan olettaa, että käsiteltävän palan data löytyy jo entuudestaan varsinaisesta deduplikointikannasta ja tähän dataan voidaan viitata osoittimella. Deduplikoinnissa datan häviäminen on siis teoriassa mahdollista, mikäli kahdella palasella on sama tiiviste. Tämän todennäköisyys riippuu täysin valituista algoritmeista ja tiivisteiden pituuksista, mutta luonnollisesti eri toteutuksissa pyritään minimoimaan datan häviämisen mahdollisuus. Kuva 3.4 havainnollistaa yksinkertaistetun kuvan avulla deduplikoinnin periaatteen, jossa nuolet esittävät osoittimia ja kuvan alareunassa olevat vihreät laatikot muodostavat deduplikointikannan.



Kuva 3.4. *Deduplikointia havainnollistava piirros.*

Yksinkertaisimmillaan deduplikointi voidaan toteuttaa kiinteämittaisilla palasilla, jolloin toimenpide on luonnollisesti yksinkertaisempi ja vaatii täten vähemmän prosessointiresursseja deduplikointijärjestelmältä. Parempaan pakkaussuhteeseen vaaditaan sen sijaan kehittyneempiä menetelmiä, kuten esimerkiksi liukuvaan ikkunaan perustuvaan datan paloittelua, jossa palojen rajat valitaan datavirrasta siten, että palaset ovat mahdollisimman suuria ja niiden löytyminen ei rajoitu ennalta määritettyjen rajojen sijaintiin (Faritha Banu & Chandrasekar 2012, s. 364.). Kuva 3.5 selventää liukuvan ikkunan (ylempi) ja kiinteämittaisten palasten (alempi) toteutusten eroa. Deduplikoituva data on tässä kuvattu merkkijonolla ABC.



Kuva 3.5. *Deduplikoinnin liukuvaa ikkunaa havainnollistava piirros.*

Deduplikointi vaatii prosessointitehon lisäksi myös nopeaa muistia indeksiä varten. Indeksiin tallennetaan palasten tiivisteet siten, että data voidaan näiden avulla löytää varsinaiselta deduplikoidun datan tallentamiseen käytettävältä levytilalta. Tämä indeksi säilötään levyä nopeampaan RAM-muistiin (Random Access Memory) ja mahdollinen RAM-muistin loppuminen aiheuttaa indeksi sivuttamisen (engl. paging) hitaammalle levylle heikentäen koko järjestelmän suorituskyvyn. Erittäin suurien deduplikoitujen tietomassojen indeksointiin voidaan käyttää kahdessa kerroksessa olevaa indeksiä, joista toinen mahtuu käytettävissä olevaan RAM-muistiin ja toinen kerros voi sijaita hitaammalla levypohjaisella medialla. (Bhagwat et al. 2009) Tähän ei kuitenkaan aina ole tarvetta. Symantec ohjeistaa omaan Backup Exec 2012 varmistusjärjestelmän levypohjaiseen deduplikointiin varattavaksi 1,5 gigatavua RAM-muistia jokaista teratavua deduplikoitua dataa kohti (Symantec 2012, s. 756.).

Täten esimerkiksi varsin maltillinen 24 gigatavua muistia sisältävä varmistuspalvelin mahdollistaa 16 teratavun deduplikointikannan käyttämisen. Mikäli pakkaussuhteeksi oletetaan 1:10, saavutetaan tällä 160 teratavun bruttokapasiteetti, jonka saavuttaminen ilman deduplikointia olisi monin verroin kalliimpaa. Samalla saavutetaan konesalissa tehokkaampi käyttöaste konesalin räkkikaapeissa ja pienempi virran kulutus, sillä fyysisiä levyjä on vähemmän.

Datan deduplikointi sopii hyvin virtuaalisten palvelinten varmistamiseen, sillä esimerkiksi monissa konesaliympäristöissä on useita saman käyttöjärjestelmän omaavia virtuaalisia palvelimia asennettuna. Tällöin esimerkiksi itse käyttöjärjestelmien omat tiedostot sisältävät paljon deduplikoituvaa dataa. Varmistusten määrittely on myös tällöin helppoa, sillä kaikista palvelimista voidaan ottaa päivittäin täydelliset varmuuskopiot, eikä inkrementaalisia tai differentiaalisia varmistuksia välttämättä tarvita ollenkaan. Tämä tosin edellyttää, että varmistettavien palvelinten ja deduplikointikannan välinen yhteys on riittävä kaistan leveydeltään. Nopeimmat SAN-verkot kykenevät 16 gigabitin nopeuteen, mutta myös gigabitin LAN-verkon yli pystyy kohtuullisen suuren määrän dataa siirtämään toimistoaikojen ulkopuolelle sijoittuvan aikaikkunan rajoissa.

Myös monet organisaatioiden käytössä olevat tietojärjestelmät sisältävät itsessään helposti deduplikoituvaa dataa. Esimerkkinä voisi mainita sähköpostipalvelimen ja tiedostopalvelimen. Varsinkin monet sähköpostin välityksellä lähetetyt tiedostot saattavat löytyä useiden käyttäjien sähköpostitileiltä, jolloin sähköpostipalvelimen kanta on hyvin deduplikoituvaa. Sen sijaan palvelimet, jotka sisältävät valmiiksi tiiviisti pakattua ja vähäredundanttista dataa, eivät saavuta korkeita deduplikointisuhteita.

4. LÄHTÖTILANNE

Tässä luvussa kuvataan lähtötilanne, eli minkälainen varmistusjärjestelmä oli asiakkaalla alun perin käytössä. Käytössä oleva muu infrastruktuuri vaikuttaa myös merkittävästi varmistusjärjestelmään, joten tästä syystä tämä on myös kuvattu pääpiirteittäin. Yksityiskohtaisimmat tiedot järjestelmästä on sen sijaan jätetty jo tietoturvallisuuden takia kertomatta. Infrastruktuurilla tarkoitetaan tässä tapauksessa palvelimia, levyjärjestelmiä ja tietoliikenneratkaisuja. Tässä luvussa kuvataan myös muutamia ongelmia, joihin vanhan järjestelmän käytön aikana oli kohdattu. Nämä ongelmat olivat perimmäinen syy järjestelmän uusimiseen.

4.1 Järjestelmäarkkitehtuuri

Järjestelmäarkkitehtuurilla viitataan tässä tapauksessa asiakkaan käytössä olevan infrastruktuurin lisäksi käytössä oleviin palveluihin.

4.1.1 Merkittävimmät palvelut

Asiakkaan käytössä olevat tietojärjestelmät perustuvat pääosin Microsoftin työasemaympäristöön sekä lukuisiin muihin sovelluspalvelimiin. Lähitulevaisuudessa liiketoiminnan kannalta merkittävimmät sovellukset ovat selainpohjaisia ja nämä sovellukset toimivat virtuaalisilla Linux-palvelimilla. Merkittävimmillä sovelluksilla tarkoitetaan tässä yhteydessä rahti- sekä lipunmyyntijärjestelmää ja matkakortin lataamiseen tarkoitettua järjestelmää.

Osa vanhemmista järjestelmistä toimii IBM:n AIX-käyttöjärjestelmää käyttävissä palvelimissa, joista asiakas on hiljalleen siirtämässä palveluita muihin ympäristöihin, jolloin käytössä olevien erilaisten käyttöjärjestelmien ja laiteympäristöjen lukumäärä saadaan vähennettyä ja siten ylläpito helpottuu.

Myös työasemaympäristöön ja sen hallintaan käytetyt Microsoft Windows Server -sarjan palvelimet, joita ovat esimerkiksi toimialueen hallintapalvelimet (Domain Controller, DC), DHCP- ja DNS-palvelimet (Dynamic Host Configuration Protocol ja Domain Name System), MS Exchange -palvelin (sähköposti), tiedosto- sekä tulostuspalvelimet, ovat kaikki virtualisoituja. Tämän lisäksi löytyy joukko muita esimerkiksi taloushallintoa varten asennettuja sovelluspalvelimia.

4.1.2 Tietojärjestelmä pääpiirteissään

Tietojärjestelmä perustuu suurelta osin asiakas−palvelin-malliin ja palvelimet ovat keskitetty asiakkaan kahteen käytössä olevaan konesaliin. Eri toimipisteillä, joita asiakkaalla on noin 50, ei siis ole juurikaan hajautettuja palvelinratkaisuja pääkonttoria ja suurinta yksittäistä toimipistettä lukuun ottamatta.

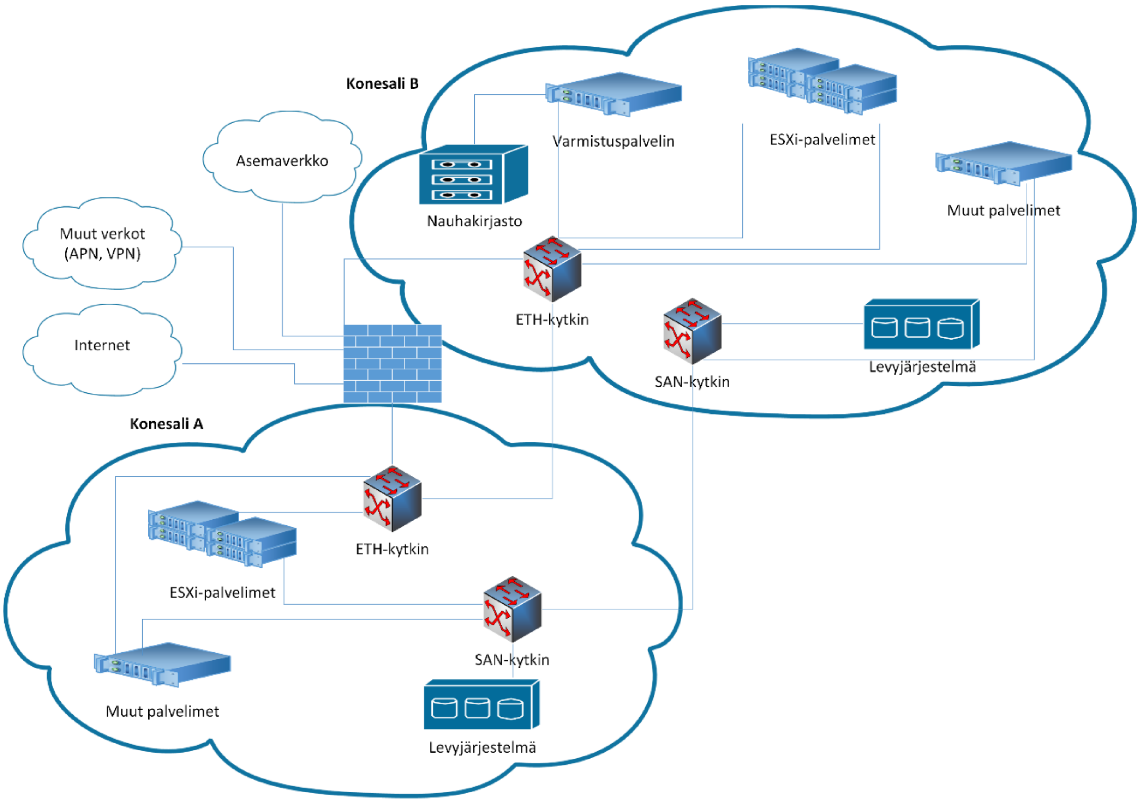
Palvelimet on hajautettu kahteen konesaliin siten, että liiketoiminnan kannalta kriittisimmät toiminnot pystytään suorittamaan ilman toista konesalia. Toinen konesaleista ei kuitenkaan ole varalla, vaan normaalissa tilanteessa kuormitus pyritään tasaamaan näiden kahden konesalien palvelinten välillä.

Palveluiden kannalta tärkeimmät järjestelmät konesaleissa ovat kaksi erillistä replikoivaa levyjärjestelmää, virtuaalipalvelinympäristö (ESXi-palvelimet), erilliset tietokantapalvelimet, sovellusvirtualisointipalvelimet (Citrix), muut fyysiset sovelluspalvelimet, palomuurit sekä LAN- ja SAN-verkot. Erilaisten virtualisointiratkaisujen käytöllä on siis selkeästi pyritty tekemään järjestelmästä helposti skaalautuva, sillä esimerkiksi levyjärjestelmän avulla pystytään palvelimille helposti provisioimaan täsmälleen haluttu määrä levytilaa, jolloin levyjen käyttöaste saadaan korkeaksi ja ennen kaikkea uuden palvelimen käyttöönotto erittäin nopeaksi. Yhtä lailla virtuaalipalvelinympäristö tukee edellä mainittuja tavoitteita.

Kuvassa 4.1 on esitetty konesalien rakenne pääpiirteittäin. Yksinkertaistuksen vuoksi kuvaan ei ole piirretty esimerkiksi tietokantapalvelimia erikseen, sillä pohjimmiltaan ne ovat aivan tavallisia fyysisiä palvelimia, kuten myös ESXi-ympäristön yksittäiset isäntäkoneet. ESXi-palvelinten roolia on tässä vain haluttu korostaa.

4.1.3 Varmistusjärjestelmä pääpiirteissään

Asiakkaan alkuperäinen varmistusjärjestelmä sisälsi vain fyysisen Windows-palvelimen sekä 24 nauhan nauhakirjaston ja kummatkin olivat sijoitettu samaan konesaliin. Itse palvelimessa oli levytilaa vain käyttöjärjestelmää ja ohjelmistoja varten, mutta väliaikaista datan tallennusta (D2D2T) varten oli levyjärjestelmästä osoitettu tarvittava määrä levytilaa, jotta kyettiin ajamaan useampaa varmistustyötä (engl. backup job) samanaikaisesti ja näin ollen lyhentämään varmistuksiin käytettyä aikaikkunaa. Levylle siirretty data voitiin rauhassa siirtää vaikka keskellä päivää nauhakirjastoon tuotantojärjestelmien häiriintymättä, sillä vain yksi varmistustyö pystyi kirjoittamaan kerrallaan dataa nauhalle, sillä nauhakirjasto sisälsi vain yhden nauhurin.



Kuva 4.1. *Asiakkaan konesalien järjestelmät pääpiirteittäin.*

4.2 Tietoliikenne

Konesalien väliset verkot ovat toteutettu valokuidulla, niin kutsutulla mustalla kuidulla, joka mahdollistaa nopean SAN-verkon toteuttamisen. Riittävän nopea verkko on myös edellytys toimivalle levyjärjestelmien väliselle reaaliaikaiselle replikoinnille. Käytössä on kytketty SAN-verkko, jossa liityntätapana on 4 gigabitin FC. Kaikki SAN-verkon laitteet ovat kytketty konesalissa olevaan kytkimeen lukuun ottamatta nauhakirjastoa, joka on kytketty suoraan sitä ohjaavaan palvelimeen.

Yhteydet julkiseen internetiin, asiakkaan eri toimipisteiden MPLS-tekniikkaa (Multiprotocol Label Switching) hyödyntävät WAN-verkot sekä muut operaattorin toteuttamat verkkoratkaisut, joita ovat esimerkiksi VPN-yhteydet (Virtual Private Network) ja yksityiset (engl. custom tai private) APN-verkot (Access Point Name) terminoidaan konesaleissa olevien palomuurien kautta osaksi suurempaa verkkokokonaisuutta. Liikennettä valvoo asiakkaan tiloissa olevat, mutta operaattorin ylläpitämät palomuurit. Erityisesti huomioitavaa on se, että asemien käytössä oleva WAN-verkko perustuu pääosin DSL-tekniikkaan (Digital Line Subscriber) ja hitaimmat yhteydet ovat vain yhden megabitin nopeuteen kykeneviä. Tämän takia esimerkiksi asemilla olevia työasemia ei varmuuskopioida verkon yli konesaliin.

4.3 Järjestelmän päivittämiseen johtaneet syyt

Vanhalla varmistusjärjestelmällä alkoi olla jo varsin paljon ikää ja tekniikka vanhahtavaa. Järjestelmän uusimiseksi oli kuitenkin monia syitä, joihin pureudutaan seuraavissa luvuissa. Itse laitteiston ikä oli yksi suurimmista motivaattoreista järjestelmän uusinnan kannalta. Esimerkiksi nauhakirjastoon liittyvä nauhuri oli ollut laitteen lokien mukaan yhteensä melkein 25 000 tuntia päällä, joten vikaantumisen todennäköisyys kasvoi ajan myötä. Myös itse varmistuspalvelimen huoltosopimus oli loppumassa, joskin tätä olisi voinut myös jatkaa. Huomioitavaa on, että vanhempien laitteiden huoltosopimusten hinnat ovat pääsääntöisesti uudempia laitteita korkeammat.

4.3.1 Kaksi eri järjestelmää

Varmistuspalvelimelle oli asennettu kaksi erillistä ohjelmistoa, jotka ottivat hieman eri tavalla varmuuskopioita. Toinen ohjelmista varmuuskopioi määritetyistä virtuaalipalvelimista niiden näköistiedostot (engl. image). Näistä varmuuskopioista ei yksittäisen tiedoston palauttaminen ole kovinkaan käytännöllistä, sillä se vaatii ensin kyseisen näköistiedoston palauttamisen uudeksi virtuaalipalvelimeksi, jonka jälkeen pääsee vasta varmistetun virtuaalipalvelimen tiedostoihin käsiksi. Toisaalta kokonaisen virtuaalipalvelimen palauttaminen oli sinällään varsin suoraviivainen toimenpide. Esimerkiksi tiedostopalvelimen osalta kokonaisen virtuaalipalvelimen palauttaminen vanhan päälle ei ole edes tarkoituksenmukaista, sillä tällöin kaikki käyttäjät joutuisivat palamaan tiedostopalvelimella olevien tiedostojen osalta edellisen päivän tilanteeseen.

Tiedostotason varmistuksia varten oli puolestaan käytössä kokonaan erillinen ohjelmisto samalla varmistuspalvelimella. Tämän avulla pystyi erikseen määrittämään varmistettavat hakemistot, johon ei näköistiedoston varmuuskopioinnissa ole mahdollisuutta. Tämä varmistustapa vaatii palvelimelle asennettavan agenttisovelluksen, joka kommunikoi varsinaisen varmistuspalvelimen kanssa. Agentit sai asennettua myös fyysisille palvelimille, jolloin myös niistä oli mahdollista ottaa tiedostotason varmistukset. Näköistiedostojen varmuuskopiointi ei vaatinut virtuaalipalvelimille mitään erikseen asennettavia sovelluksia, vaan ainoastaan virtuaaliympäristöä hallitsevaan sovellukseen lisäosan. Tämän ansiosta virtuaaliympäristön varmistaminen onkin monesti huomattavasti helpompaa ja puoltaa vahvasti virtuaalipalvelinten käyttöä.

Tästä kahden päällekkäisen varmistusjärjestelmän käytöstä haluttiin kuitenkin luopua, sillä monissa uudemmissa varmistusjärjestelmissä oli mahdollisuus suorittaa sekä näköistiedostojen että tiedostotason varmistuksia samalla sovelluksella ja peräti samanaikaisesti.

4.3.2 Varmuuskopioinnin kesto

Varmistusten käyttämät aikaikkunat olivat myös kasvaneet varsin pitkiksi, mikä vaikeutti myös osaltaan datan palauttamista. Esimerkiksi eräs päivittäin suoritettu inkrementaalinen varmistustyö saattoi kestää melkein 11 tuntia. Tämän lisäksi otettiin päivittäin nauhalle vielä suurimmasta osasta virtuaalipalvelimia levyn näköistiedostot, jotka tosin olivat ensin kirjoitettu levylle (D2D2T).

Varmistustöiden ei tulisi olla käynnissä silloin, kun itse järjestelmän kuormitus on suurimmillaan. Tässä tapauksessa tuotantojärjestelmien kuormitus keskittyy varsin selkeästi toimipisteiden aukioloaikojen mukaiselle ajanjaksolle, joten varmistusaikaikkuna on sovitettava tämän ulkopuolelle.

4.3.3 Nauhalta palauttamisen hitaus

Pelkästään nauhalle perustuvaan varmistetun datan palauttaminen saattoi tilanteesta riippuen olla erittäin hidasta. Tätä ei myöskään helpottanut ollenkaan pitkäksi venyvät varmistustyöt, jotka varasivat nauhurin käyttöönsä, jolloin esimerkiksi yksittäisen tiedoston palauttamiseksi oli joko odotettava käynnissä olleen varmistustyön loppumista tai keskeytettävä se. Yksittäisen tiedoston palauttamiseen kului myös jonkin verran aikaa, kun nauhakirjasto siirsi oikean nauhan nauhuriin ja etsi nauhalta kohdan, josta haluttu data löytyi.

Yksittäisen pienen tiedoston palauttaminen oli siis esimerkiksi kokonaisen virtuaalikoneen palauttamiseen nähden erittäin hidasta, mikäli vertailtiin palautettavan datan määrää ja tähän kuluvaa aikaa. Mittaustulosten esittäminen on kuitenkin ongelmallista, sillä palauttamisen nopeus on erittäin tapauskohtaista ja se riippuu monesta eri tekijästä. Näitä ovat esimerkiksi palautettavan palvelimen resurssit sekä palautettavan datan ominaisuudet, eli esimerkiksi tiedostojen koko ja lukumäärä. Useamman kuukauden ikäisen tiedoston palauttaminen muuttaa tilannetta vielä siten, että haluttu nauha ei ole enää nauhakirjastossa, vaan tämä on tuotu kuukausinauhana ulos konesalista. Näin vanhan tiedoston palauttaminen vaatii siis vielä käynnin konesalilla ja oikean nauhan syöttämisen nauhakirjastoon tiedoston palauttamista varten.

4.4 Uuden järjestelmän keskeiset tavoitteet

Keskeisimpinä tavoitteena oli nykyaikaistaa varmistusjärjestelmä teknisesti niin, että se tukisi myös tulevia virtuaaliympäristön tarjoamia teknologioita ja mahdollistaisi koko infrastruktuurin pitämisen ajan tasalla käyttämällä valmistajan tukemia ohjelmistoversioita. Kunkin sovelluksen elinkaari on suunniteltava niin, että sen alustana voidaan käyttää tuettua käyttöjärjestelmäversiota.

Uuden järjestelmän tavoitteena oli myös vähentää ylläpitoon käytettävää työmäärää. Tämä tarkoitti siis siirtymistä kahdesta erillisestä varmistusjärjestelmästä ratkaisuun, jossa yksi varmistusjärjestelmä kykenee tarjoamaan tarvittavat ominaisuudet.

Olennaista oli myös saada varmistuksiin ja palautuksiin kuluva aika lyhentymään. Tämä tarkoitti käytännössä siirtymistä täysin nauhaan perustuvasta varmistusjärjestelmästä osittaiseen levypohjaiseen järjestelmään.

5. UUSI JÄRJESTELMÄ

Tässä luvussa kuvataan lyhyesti uuden järjestelmän hankkiminen, implementoinnissa havaitut haasteet, järjestelmän arkkitehtuuri sekä muutokset aikaisempaan järjestelmään.

5.1 Järjestelmän tekniikka

Varmistusjärjestelmä koostuu tavanomaisesta räkkiasennettavasta palvelimesta sekä tähän liitettävästi nauhakirjastosta. Järjestelmän asentamisessa käytettiin apuna yhteistyökumppanin konsulttia, jolla oli runsaasti kokemusta vastaavien järjestelmien asentamisessa ja täten asennusvaihetta saatiin nopeutettua merkittävästi. Palvelimelle asennettiin käyttöjärjestelmäksi Windows Server 2008 R2 ja itse varmistusohjelmistoksi Symantec Backup Exec 2012.

Palvelimeen asennettiin sisäisesti yhteensä 25 levyä, joista kaksi on dedikoitu järjestelmälevyiksi ja lopuista 23 levystä tehtiin kaksi eri loogista levyasemaa levytallennuksia varten RAID5-tekniikalla (Redundant Array of Independent Disks), siten että yksi levy toimii varalevynä (engl. hot spare). Varalevyn käyttö oli perusteltua, sillä vaikka RAID5-tasoinen levyjärjestelmä tarjoaakin vikasietoisuuden yksittäisen levyn rikkoutumista vastaan, saattaa varmistuspalvelimeen sisäänrakennetun levyjärjestelmän rekonstruoinnissa kestää pitkään, eikä rikkoutunutta levyä välttämättä saada välittömästi korvattua uudella. RAID-pakan korruptoituminen saattaisi tässä tapauksessa aiheuttaa viimeisen kuukauden ajalta päivittäisten varmistusten menettämisen, joten tästä syystä vikasietoisuuteen kiinnitettiin erityistä huomiota.

Varmistuspalvelin sisälsi kaksi neliytimistä 2,4 gigahertsin kellotaajuudella toimivaa Intel Xeon E5-2609 suoritinta sekä yhteensä 32 gigatavua muistia, joka oli jaettu tasan kummankin suorittimen kesken.

Nauhakirjasto kytkettiin suoraan varmistuspalvelimeen kuidulla (FC), sillä tarvetta monipuolisemman SAN-verkon rakentamiseksi ei ollut ja tällöin kytkennästä saatiin yksinkertaisempi. Mahdollista olisi myös ollut kytkeä varmistusjärjestelmä suoraan SAN-verkkoon ja ajaa varmistukset levyjärjestelmistä tätä yhteyttä käyttäen. Tätä ei kuitenkaan toteutettu konsultin suosituksesta, sillä mahdollinen virheellinen konfigurointi varmistusjärjestelmän käyttöjärjestelmässä saattaisi korruptoida levyjärjestelmän tuhoten koko virtuaalipalvelinympäristön.

5.2 Muutokset aikaisempaan

Vanha järjestelmä perustui fyysisesti pienempään korttipalvelimeen, jossa sisäistä levytilaa oli vain järjestelmälevyä varten. Korttipalvelimen käyttö ei ollut tässä tapauksessa mahdollista, sillä itse palvelimeen haluttiin sisäänrakennettuna riittävästi levytilaa levyvarmistuksia varten. Levyjärjestelmän käyttö ei olisi ollut järkevää, sillä koko varmistusjärjestelmä haluttiin muusta käytössä olevasta järjestelmästä riippumattomaksi. Tässä tapauksessa riippumattomuus on viety niin pitkälle, että varmistusjärjestelmä on yhteydessä muuhun järjestelmään vain lähiverkkoyhteyden osalta.

5.3 Käytössä havaitut haasteet

Varmistusjärjestelmän käytön aikana havaittiin erilaisia haasteita, joista osa on jäänyt selittymättömäksi yhteensopivuusongelmiksi eri järjestelmien välillä. Ongelmia jouduttiin muutamassa tapauksessa selvittelemään myös ohjelmiston valmistajan tuen kanssa. Erään ohjelmistovirheen seurauksena järjestelmä antoi tunnin välein virheilmoituksen puuttuvasta lisenssistä, johon liittyvää ominaisuutta ei järjestelmässä kuitenkaan ollut käytössä. Tämä ongelma ratkesi vasta kolmannen ohjelmistolle julkaistun laajemman korjauspäivityksen (engl. service pack) myötä.

Myös uuden järjestelmän käyttöliittymä ja -logiikka oli muuttunut aikaisemmista versioista, eikä tämä saanut käyttäjiltä täysin yksimielistä vastaanottoa (Symantec Community 2012). Seuraavassa kahdessa alaluvussa on kuitenkin keskitytty kahteen konkreettisempaan ongelmaan ja niiden ratkaisuihin.

5.3.1 Pienten tiedostojen vaikutus suorituskykyyn

Fyysisten tietokantapalvelinten varmistustöissä havaittiin, että näiden aikaikkuna alkoi venyä, vaikka varmistettavan datan määrä ei kasvanutkaan samassa tahdissa. Selkeää syytä tähän ongelmaan ei aluksi löytynyt. Tietokantapalvelimen täyden viikoittaisen tai kuukausittaisen varmistuksen suorittamisessa kului lopulta vaihtelevasti noin seitsemästä tunnista aina melkein kymmeneen tuntiin, vaikka varmistettavan datan määrä vaihteli vain noin 133 ja 165 gigatavun välillä. Tämä oli ehdottomasti liian pitkä aika yhdelle varmistustyölle, sillä tuona aikana oli nauhakirjaston nauhuri varattuna ja muut nauhalle kirjoitettavat varmistustyöt odottivat tämän valmistumista. Koska tietokantaklusterissa oli myös toinen samanlainen palvelin, saattoi viikonloppuisin klusterin varmistuksissa kestää yhteensä lähes vuorokausi.

Varmistustyön hitauteen löytyi lopulta varsin selkeä syy. Tietokantapalvelimen ohjelmisto oli järjestelmän käyttöönotosta alkaen kerännyt auditointilokia. Tämä auditointiloki koostui yli 1,2 miljoonasta pienestä tiedostosta, joiden yhteenlaskettu koko oli tosin vain noin 4,9 gigatavua. Tiedostojen lukumäärä laskettiin Linuxin find- ja wc-komentoa (word count) käyttäen:

```
# cd /opt/oracle/11.2.0/grid/rdbms/
# find audit/ -print | wc –l
1263518
```

Tässä putkitetussa komennossa *find* tulostaa (-print) ensimmäisenä parametrina annetusta hakemistosta listauksen tiedostoista ja komento *wc* laskee parametrin *–l* (--lines) mukaisesti tulostettujen rivien määrän. Tässä tapauksessa siis kyseisessä hakemistossa olevien auditointitiedostojen lukumäärän.

Tähän lokiin oli kerätty dataa kaikkien sellaisten käyttäjien toimenpiteistä, joiden käyttöoikeustaso on joko SYSDBA tai SYSOPER. Ohjelmiston valmistaja on määrittänyt pakolliseksi (engl. mandatory auditing) tämän lokituksen (Oracle). Kyseinen tietokantapalvelin oli myös otettu käyttöön suunnilleen samaan aikaan kuin itse varmistusjärjestelmä, joten aivan aluksi auditointilokin määrä oli varsin rajallinen, eikä täten vaikuttanut järjestelmän suorituskykyyn merkittävästi.

Varmistustöiden suorittamisesta ei lokia varmistuspalvelimelle ole järjestelmän käyttöönotosta saakka säilytetty, mutta tietokantapalvelinten varmistustöiden käyttämä aika putosi melko dramaattisesti, kun nämä auditointitiedostoja sisältävät hakemistot jätettiin varmistusten ulkopuolelle.

Ennen noin kymmenen tuntia nauhalle kirjoitettu varmistustyö vei enää alle tunnin. Myös päivittäin ajettavat inkrementaaliset deduplikointikantaan ajetut varmistukset kestivät aikaisemman noin kuuden tunnin sijasta vain noin puoli tuntia. Deduplikointikantaan ajetuissa varmistuksissa on tosin huomattava, että samaan aikaan oli käynnissä myös muita varmistustöitä, jolloin varmistettava palvelin joutui jakamaan datan siirtoon käytettävän gigabitin Ethernet-verkon (kaksi aggregoitua gigabitin verkkoliitäntää) muiden käynnissä olleiden varmistustöiden kanssa.

Taulukoon 5.1 on listattu kahdelta eri viikolta eri varmistustöiden suorittamiseen kuluneet ajat, siirretyn datan määrä sekä nopeus. Tiedot on kerätty siten, että näiden tarkastelujaksojen välistä on jätetty kolmen viikon tulokset pois. Tuona aikana tulokset vaihtelivat huomattavasti, sillä ennen ongelman juurisyyn löytymistä kokeiltiin muutamia erilaisia varmistustöiden määrityksiä ongelman poissulkemiseksi. Numeeriset tulokset on poimittu varmistusjärjestelmän raportoimista lokitiedoista. Taulukon luettavuuden helpottamiseksi on erityyppisistä varmistustöistä käytetty lyhenteitä. Lyhenteiden tarkoitukset ovat seuraavat:

- Incr. daily dedup: arkipäivisin deduplikointikantaan ajettu inkrementaalinen varmistustyö, vain edellisen varmistustyön jälkeen muuttuneet varmistettavat tiedostot on varmistettu

- Full dedup: kerran viikossa deduplikointikantaan ajettu täydellinen varmistustyö, kaikki varmistettavat tiedostot
- Full monthly tape: neljän viikon välein nauhalle ajettu täydellinen varmistustyö, kaikki varmistettavat tiedostot

Taulukko 5.1. *Varmistustöihin kulunut aika, siirretyn datan määrä ja varmistusten nopeus kahden vertailtavan viikon ajalta.*

Varmistustyö	Kesto	Datan määrä (Gt)	Nopeus (Mt/min)
Incr. daily dedup.	3 h 54 min	24	109
Incr. daily dedup.	3 h 52 min	24	107
Incr. daily dedup.	4 h 20 min	25	103
Incr. daily dedup.	4 h 7 min	24	101
Incr. daily dedup.	4 h 48 min	59	232
Full dedup.	8 h 43 min	140	326
Full monthly tape	7 h 31 min	137	322
Incr. daily dedup.	11 min	21	4436
Incr. daily dedup.	18 min	35	4380
Incr. daily dedup.	39 min	22	1061
Incr. daily dedup.	30 min	22	1335
Incr. daily dedup.	51 min	29	1138
Full dedup.	2 h 34 min	143	1327
Full monthly tape	53 min	141	4132

Tuloksista käy selvästi ilmi pienten tiedostojen vaikutus järjestelmän suorituskykyyn. On kuitenkin huomattava, että mittauksia ei ole tehty eristetyssä laboratorioympäristössä, sillä tulokset on poimittu todellisesta tuotantoympäristöstä. Muut järjestelmät vaikuttavat aina jonkin verran yksittäisten varmistustöiden suorittamiseen, eli tuloksista ei ole pystytty täysin poissulkemaan ulkoisia tekijöitä. Esimerkiksi kaikkien deduplikointikantaan ajettujen töiden aikana on ollut muita varmistustöitä käynnissä. Sen sijaan nauhavarmistuksen aikaan ei muita töitä ole ollut käynnissä.

Edellä mainitut suorituskykyparannukset saatiin aikaiseksi jättämällä Oraclen tietokantapalvelimen auditointitiedostot pois varmistustyöstä. Itse tiedostot piti myös saada palvelimilta säännöllisesti poistumaan, jotta vältyttäisiin mahdollisilta tiedostojärjestelmän ongelmilta. Suurten tiedostomäärien käsittely ei kuitenkaan osoittautunut aivan suoraviivaiseksi toimenpiteeksi. Yli kuukauden vanhoja tiedostoja yritettiin ensin poistaa käyttäen jälleen find-komentoa.

```
# find /opt/oracle/11.2.0/grid/rdbms/audit/* -mtime +31 -exec rm {} \;
-bash: /bin/find: Argument list too long
```

Tämä ei kuitenkaan toiminut, vaan komentotulkki antoi virheen, vaikka tiedostoja yritettiin siivota pienemmissä erissä, eli suurentamalla parametrin *-mtime* arvoa. Komennon *find* parametri *-exec* suorittaa komennon *rm* (remove). Poistokomento saa muuttujan *{}* avulla kaikki komennon *find* löytämät tiedostot ja *\;* päättää komennon *exec*. Ongelma saatiin kierrettyä muuttamalla annettua komentoa hieman, mutta komentotulkin antamaan virheilmoitukseen ja sen aiheuttajaan ei tässä tarkemmin pureuduta.

```
# find /opt/oracle/11.2.0/grid/rdbms/audit/ -type f -mtime +31 -exec rm {} \;
```

Tällöin poisto onnistui, mutta poisto ajettiin edelleenkin pienemmissä erissä, sillä tuotantokäytössä olevaa palvelinta ei haluttu kuormittaa liikaa.

Tämä tapaus osoitti selkeästi, että varmistusjärjestelmän suorituskykyyn vaikuttaa merkittävästi varmistettavan datan tyyppi, eikä varsinkaan suorituskykyongelmia voi aina ratkaista kasvattamalla esimerkiksi käytettävissä olevaa kaistaa. Kyseisessä tapauksessa myös merkille pantavaa on se, että varmistusten ylläpito linkittyy tiiviisti itse varmistettaviin järjestelmiin ja vaatii siten myös näiden järjestelmien osaamista tai yhteistyötä eri järjestelmien ylläpitäjien välillä. On myös huomattavaa, että saman datan varmistaminen levykuvana ei olisi vastaavanlaista suorituskykyongelmaa aiheuttanut, mutta levykuvasta yksittäisten tiedostojen palauttaminen on huomattavasti monimutkaisempi toimenpide, eikä käytössä oleva varmistusjärjestelmä mahdollista fyysisen palvelimen varmistamista levykuvana, vaan ainoastaan tiedostotasolla. Virtuaalipalvelimia ei tämä rajoitus tosin koske.

5.3.2 Deduplikointikannan korruptoituminen

Varmistusjärjestelmän oltua käytössä noin yhdeksän kuukautta huomattiin erään viikonlopun jälkeen kaikkien deduplikointikantaan ajettujen varmistustöiden (engl. backup job) olleen edelleen käynnissä, mutta dataa nämä työt eivät olleet saaneet varmistettua lokitiedostojen mukaan yhtään. Ensiavuksi tähän ongelmaan yritettiin varmistuspalvelimen uudelleenkäynnistystä sekä joukkoa muita varmistusohjelmiston valmistajan tukifoorumeilta löytyviä keinoja.

Varsinaisesti tukipyyntöä ongelmasta ei luotu, sillä ongelmaan pyrittiin saamaan nopeasti korjaus. Tukipyynnön avulla varmistusjärjestelmään liittyvien ongelmien ratkaisuista oli aikaisempia kokemuksia, eivätkä vasteajat olleet tällöin erityisen nopeita. Osittain tähän vaikuttaa toisella mantereella sijaitsevan tukipalvelun aikaero. Korruptoituneen kannan korjaamisen mahdollisuudesta ei ollut myöskään minkäänlaisia takeita.

Deduplikointikannan ongelmien selvittelyn aikana otettiin tuotannon kannalta kriittisimmistä järjestelmistä varmuuskopiot normaalista poiketen päivittäin nauhalle ja levylle ilman deduplikointia. Tällä mahdollistettiin kriittisten tietojen palauttaminen poikkeustilanteessakin.

Itse deduplikointikannasta oli myös olemassa nauhalle otettu varmuuskopio, mutta tämän koko oli noin yhdeksän teratavua, joten vanhan kannan palauttamista ei edes yritetty. Syitä tähän oli kaksi. Deduplikointikannan palauttamiseen olisi kestänyt kauan ja nauhalla olevan datan tilasta ei ollut tässä vaiheessa varmuutta, sillä deduplikointikanta on voinut sisältää jo pidemmän ajan korruptoitunutta dataa.

Deduplikointikanta sisälsi tuolloin yli 110 000 tiedostoa ja oli kooltaan noin yhdeksän teratavua. Nauhalta depuplikointikannan dataa oli suoritetuissa testeissä palautettu varmistuspalvelimen paikalliselle levylle noin yhdeksän gigatavun minuuttinopeudella. Datan palauttamisessa edes Ethernet-verkon nopeus ei ollut rajoittava tekijä, sillä nauhakirjaston LTO-5-nauhuri oli kytketty FC-yhteydellä suoraan varmistuspalvelimeen, jossa järjestelmän levyt olivat paikallisesti asennettuina. Tämä olisi tarkoittanut palautustyöhön kuluvaksi ajaksi yhdeksän teratavun kannalle

$$t = \frac{9000\ Gt}{9\frac{Gt}{min}} = 1000\ min, \tag{5.1}$$

eli noin 17 tuntia, jonka aikana ei luonnollisesti datan palautukseen käytettävää nauhuria olisi voinut käyttää muiden varmistusten ottamiseen. Itse deduplikointikannan palautusoperaatiosta ei ollut myöskään aiempaa kokemusta ja deduplikointikanta ei ollut ainoa varmistukseen käytetty media, joten näiden perusteella päätettiin vanha korruptoitunut deduplikointikanta poistaa kokonaan ja luoda tämä uudestaan.

Deduplikointikannan uudelleen luomisen jälkeen varmistustyöt toimivat jälleen normaalisti ja varmistustöiden kestot vakiintuivat nopeasti. Tapaus osoitti, että järjestelmä vaatii säännöllistä valvontaa ja varmistustöiden määrityksiin tulee kiinnittää huomiota. Esimerkiksi varmistustyölle tulee määrittää maksimiaika, jonka se saa olla käynnissä ja tämän ajan ylittyessä tulee tästä lähettää automaattinen raportti järjestelmän ylläpidolle. Kannan korruptoituminen oli helposti havaittavissa, sillä yksikään deduplikointikantaan ajettu varmistustyö ei toiminut, eikä kannasta saanut myöskään mitään dataa palautettua. Varsinaista syytä korruptoitumiselle ei löytynyt.

6. MITTAUSTULOKSIA

Tässä luvussa on esitetty kvantitatiivista dataa kahdesta eri varmistusjärjestelmästä ja pyritty vertailemaan näitä keskenään. Kuten aiemmin on jo mainittu, vertailu ei ole kovin eksaktia, sillä mittaustulokset eivät ole toteutettu eristetyissä laboratorio-olosuhteissa, vaan ne ovat kerätty tuotantokäytössä olevien järjestelmien lokeista. Järjestelmien toimintaperiaatteet eroavat myös paljon toisistaan. Järjestelmien tapa kerätä lokia ei myöskään helpottanut vertailun tekemistä, sillä esimerkiksi vanhassa järjestelmässä yhden palvelimen eri loogiset levyosiot käsiteltiin ikään kuin omina varmistusosioinaan (engl. backup set). Levyosioiden lisäksi esimerkiksi Windows-palvelimista järjestelmä saa varmistettua myös palvelimen tilan (engl. system state) sekä Volume Shadow Copy Service -toiminnon (VSS) avulla järjestelmässä käytössä olevat tiedostot.

Lisäksi varsinkin vanhassa järjestelmässä päivittäin suoritetut inkrementaaliset varmistukset koostuivat itse asiassa neljästä erillisestä osittain toisistaan riippuvasta varmistustyöstä, sillä esimerkiksi määrätyistä palvelimista inkrementaaliset varmistukset olivat osoittautuneet niin hitaiksi, että ne kannatti ensin erillisessä varmistustyössä tallentaa levylle (B2D). Tämän takia näiden varmistusten lokien tilastojen tulkitseminen olisi muodostunut myös varsin hankalaksi. Inkrementaalisten tai differentiaalisten varmistusten lokitiedostoista poimitut tilastot eivät myöskään anna kunnollista kuvaa koko järjestelmän suorituskyvystä ja on erittäin riippuvainen varmistettavan datan luonteesta sekä varmistettavasta ympäristöstä.

Uuden ja vanhan järjestelmän teknisten erojen lisäksi itse varmistustöiden logiikka erosi merkittävästi toisistaan. Vanhalla järjestelmällä keskeisessä asemassa olivat erilaiset valintalistat (engl. selection list) ja politiikat (engl. policy), joiden avulla eri palvelimet pystyttiin määrittämään varmistettavaksi yhdellä varmistustyöllä. Uusi varmistusjärjestelmä on sen sijaan palvelinkeskeinen ja vain virtuaalisista palvelimista pystyi yhdellä varmistustyöllä ajamaan varmuuskopiot useammasta palvelimesta. Tämäkin edellytti sitä, että kaikki virtuaalipalvelimet olivat kokonaisuudessaan valittu varmistettavaksi, ja että kaikkiin sovellettiin samoja asetuksia. Uuden varmistusohjelman toimintalogiikka aiheutti sen, että eri varmistustöiden lukumäärä saattoi kasvaa merkittävästi ja täten vaikeuttaa varmistusten hallintaa. Logiikka sai myös runsaasti vastustusta valmistajan omilla tukisivustoilla (Symantec Community 2012). Tätä diplomityötä kirjoitettaessa ohjelmiston valmistaja on julkaissut uuden version, jossa on ikään kuin palattu askel taaksepäin ja annettu järjestelmän ylläpitäjille mahdollisuus kerätä useiden palvelinten varmistustyöt yhteen (Symantec White Paper 2014). Tätä versiota ohjelmistosta ei kuitenkaan vielä tätä kirjoitettaessa ole asiakkaan ympäristössä otettu käyttöön.

6.1 Varmistuksiin kuluva aika

Varmistuksiin kuluvalla ajalla on erityisesti merkitystä arkisin suoritetuissa varmistuksissa, joissa aikaikkuna on rajallinen. Viikonloppuna suoritettavalla varmistustyöllä on huomattavasti pidempi aikaikkuna käytettävissä kuin arkisin ajettavalla työllä. Tämän takia onkin luontevaa ottaa juuri viikonloppuina täydet varmistukset ja arkisin lyhentää aikaikkunaa käyttämällä inkrementaalisia tai differentiaalisia varmistuksia.

6.1.1 Vanha järjestelmä

Taulukoon 6.1 on kerätty yhteenvedot vanhan järjestelmän yhdeksästä täydestä viikoittaisesta (engl. full weekly) varmistustyöstä varmistettavan datan määrä, koko varmistustyön kesto ja näistä laskettu keskimääräinen nopeus. Samoja yhdeksää varmistustyötä on käytetty myös myöhemmin luvussa 6.2.1, jossa on tarkasteltu hieman tarkemmin kahden yksittäisen palvelimen varmistusnopeutta.

Taulukko 6.1. *Vanhan varmistusjärjestelmän viikoittaisten täysien varmistustöiden tilasto.*

	Varmistettu data (Gt)	Kesto	Nopeus (Mt/min)
1	1412	13 h 1 min	1808
2	1423	12 h 7 min	1958
3	1424	12 h 5 min	1964
4	1446	13 h 2 min	1850
5	1437	12 h 25 min	1928
6	1434	13 h 26 min	1779
7	1398	14 h 0 min	1665
8	1389	13 h 12 min	1754
9	1386	16 h 30 min	1400
Keskiarvo	1417	13 h 19 min	1789

Taulukossa 6.1 esitetyt yhdeksän viikoittaista varmistustyötä koostuivat kaikkien varmistettavien palvelinten tiedostotason varmistuksista. Viikoittain siis dataa varmistettiin nauhakirjastoon noin 1,4 teratavun verran ja tämän kirjoittamiseen kului keskimäärin yli 13 tuntia.

On kuitenkin huomattava, että täydellä varmistuksella ei tarkoiteta kaikkien palvelinten tai edes yksittäisen palvelimen kaikkien tiedostojen varmistamista, vaan ainoastaan kaikkien valittujen tiedostojen varmistamista. Koko virtuaaliympäristön sisältämä datamäärä oli moninkertainen täyteen viikoittaiseen varmistukseen verrattuna. Viikonlopun aikana olisi aikaa ollut varmistaa myös huomattavasti tätä suurempi määrä dataa, mutta tämä olisi samalla tarkoittanut, että nauhoja olisi tarvittu enemmän.

Viikoittaisissa varmistuksissa ylikirjoitussuoja oli kolme viikkoa, eli kolmen viikon takaiset viikoittaiset varmistukset olivat aina palautettavissa. Vastaavasti päivittäisissä varmistuksissa ylikirjoitussuoja oli määritetty kuuteen päivään. Tämä kuukausittainen varmistus vastasi muuten viikoittaista, mutta nimensä mukaisesti suoritettiin vain kerran kuukaudessa ja nämä nauhat tuotiin konesalista ulos muualle säilytettäväksi.

Käytössä olleeseen nauhakirjastoon oli asennettu LTO-3 nauhuri, jonka nauhoihin mahtuu natiivisti vain 400 gigatavua dataa, joskin pakkaamalla tätä keskimäärin noin kaksinkertainen määrä dataa (LTO). Nauhakirjastossa oli 24 nauhaa, joten pakkaamattomana koko nauhakirjaston kapasiteetti oli 9,6 teratavua. Tähän kaikkeen oli saatava mahtumaan viikoittaisten ja päivittäisten varmistusten lisäksi vähintään yksi kuukausittainen täysi varmistus. Esimerkiksi viikoittaiset nauhavarmistukset varasivat yhteensä kahdeksan nauhaa nauhakirjastosta käyttöönsä, sillä 1,4 teratavua vaati pakattunakin kaksi nauhaa. On myös huomattava, että nauhakierron logiikan kannalta ei eri viikon varmistuksia kannata jatkaa edellisen viikon nauhalle, sillä muuten tuorein nauhalla oleva data määrittelisi nauhan ylikirjoitussuojan laukeamisen.

6.1.2 Uusi järjestelmä

Täydellisen vertailukohdan hakeminen uuden järjestelmän varmistuksista ei ollut mahdollista, mutta taulukossa 6.2 on esitetty vastaavanlainen yhteenveto kymmenestä perättäisesti uudella järjestelmällä ajetusta varmistustyöstä. Kyseinen varmistustyö oli määritetty ottamaan täydet varmistukset viidestätoista Windows-palvelimesta viikoittain nauhalle ja sisälsi suurimman osan asiakkaan käytössä olevista Windows-palvelimista.

Taulukko 6.2. *Uuden varmistusjärjestelmän viikoittaisten täysien varmistustöiden tilasto.*

	Varmistettu data (Gt)	Kesto	Nopeus (Mt/min)
1	684	5 h 50 min	1953
2	683	5 h 53 min	1934
3	694	5 h 10 min	2238
4	687	4 h 57 min	2312
5	686	5 h 46 min	1983
6	690	5 h 33 min	2073
7	689	5 h 2 min	2282
8	720	5 h 30 min	2181
9	747	6 h 15 min	1992
10	708	5 h 12 min	2269
Keskiarvo	699	5 h 31 min	2122

Varmistustöiden keskimääräinen nopeus oli jonkin verran suurempi kuin vanhalla järjestelmällä, joka vaikuttaa suoraan varmistuksiin kuluvaan aikaan. Vielä merkittävämpää on kuitenkin se, että nämä uudella järjestelmällä ajetut varmistustyöt sisälsivät sekä tiedostotasoisen että näköistiedostojen varmistuksen. Tässä tapauksessa täysi varmistus sisälsi kaikkien valittujen viidentoista palvelimen tiedostot, eikä tätä ollut varmistusjärjestelmästä mahdollista edes rajata, sillä muuten näköistiedostoja ei olisi voinut varmistaa samalla varmistustyöllä.

Tämän varmistustyön ulkopuolelle piti jättää esimerkiksi asiakkaan käytössä ollut tiedostopalvelin, sillä sen yksi levyosio sisälsi miljoonia pieniä arkistoitavaksi tarkoitettuja tiedostoja, joiden varmuuskopioiminen olisi kestänyt tällä menetelmällä aivan liian kauan. Tätä kyseistä palvelinta varten piti siis määrittää erillinen varmistustyö, jossa arkistotiedostot olivat rajattu pois ja vain käyttäjien kotihakemistot sisällytettiin varmistustyön tiedostovalintoihin.

Viikoittain ajettiin myös neljä muuta vastaavanlaista useita virtuaalipalvelimia sisältävää varmistustyötä sekä muutamia erillisiä, yksittäisistä palvelimista ajettuja varmistustöitä. Näiden kaikkien mahduttaminen ajettavaksi viikonlopun aikana ei missään vaiheessa ole osoittautunut ongelmalliseksi ja mikäli aikaikkuna ei tähän riittäisi, olisi nauhakirjastoon mahdollisuus hankkia toinen nauhuri.

Sen sijaan päivittäin ajettavien varmuuskopioiden keston kanssa tuli selkeästi suurempia haasteita, sillä näillä ei ollut käytettävissä arkisin kuin toimistoaikojen ulkopuolelle jäävä aikaikkuna. Vaikka nämä varmistustyöt käyttävät mediana levypohjaista deduplikointikantaa, johon on mahdollista ajaa varmistukset useasta varmistustyöstä kerrallaan, tulee tässäkin raja vastaan. Luvussa 6.5 esitettyjen tilastojen perusteella varmistuspalvelimen prosessorin suorituskyky, muistin määrä tai levyohjaimen nopeus ei vaikuttaisi olevan pullonkaula järjestelmän suorituskyvyn kannalta. Tämän perusteella palvelin suoriutuisi suuremmankin datamäärän deduplikoinnista, mikäli verkkoyhteyden yli sitä pystyttäisiin järjestelmälle syöttämään. Teorian todentamiseksi tulisi järjestelmää testata kymmenen gigabitin nopeuteen kykenevällä verkkoliitännällä. Käytössä olevassa palvelininfrastruktuurissa vaan ei vielä ole tätä tekniikkaa tukevia verkkolaitteita käytössä.

Uuden järjestelmän myötä varmistetun datan määrä oli myös kasvanut merkittävästi, sillä deduplikointitekniikka mahdollistaa korkean datan pakkaussuhteen, eikä täten varmistusmedian kustannukset ole rajoittava tekijä. Selvää kuitenkin oli, että kaikista virtuaalipalvelimista ei ollut käytössä olevilla resursseilla mahdollista ottaa näköistiedostoja päivittäin, sillä tämän datamäärän siirtäminen verkon yli ei olisi ollut mahdollista edes yhden vuorokauden aikana. Vähemmän kriittisille palvelimille, joita ovat esimerkiksi testipalvelimet, määritettiin varmistukset otettavaksi vain kerran viikossa viikoittain kiertäville nauhoille.

Varmistusjärjestelmään olisi myös mahdollista kytkeä rinnakkain useampia palvelimia mediapalvelimiksi esimerkiksi siten, että toisessa konesalissa olisi oma mediapalvelin deduplikointikantana ja nauhakirjasto nauhavarmuuskopioita varten. Tämän laajennetun varmistusjärjestelmän varmistustöitä olisi kuitenkin mahdollista ohjata keskitetysti. Tämä tosin lisäisi varmistusjärjestelmän kustannuksia sekä toisen palvelimen hankinta- ja ylläpitokustannuksien että varmistusjärjestelmän lisenssikustannusten kasvamisen myötä.

6.2 Varmistusnopeus

Sekä vanha että uusi järjestelmä antavat mahdollisuuden tutkia hyvinkin tarkasti eri palvelinten varmistamisen nopeuksia (engl. throughput). Tästä ominaisuudesta saattaa olla erityisen paljon hyötyä, mikäli järjestelmän suorituskyvyssä havaitaan ongelmia. Ongelmaa pystytään tällöin tarkemmin rajaamaan. Kuten luvussa 5.3.1 oli havainnollistettu, varsinkin paljon pieniä tiedostoja sisältävät palvelimet saattavat laskea varmistusjärjestelmän suorituskykyä merkittävästi. Tämä koskee tosin vain, mikäli palvelimesta otetaan tiedostotasolla varmuuskopioita. Pelkkää näköistiedostoa varmistettaessa ei palvelimen sisältämistä tiedostoista ole varmistusjärjestelmällä mitään tietoa, joten näiden indeksointiin ei kulu resursseja. Toisaalta tällöin ei myöskään pystytä varmuuskopioista palauttamaan palvelimen sisältämiä yksittäisiä tiedostoja ilman, että koko palvelin palautettaisiin ensin.

Seuraaviin alalukuihin on kerätty käytännön esimerkkejä sekä vanhasta että uudesta varmistusjärjestelmästä. Identtistä vertailua ei pysty tekemään, sillä uuden varmistusjärjestelmän toiminta erosi perusteellisesti vanhasta varmistusjärjestelmästä.

6.2.1 Vanha järjestelmä

Taulukkoon 6.3 on koottu otteita yhdeksästä onnistuneesta viikoittain suoritettavasta täydestä (engl. full weekly) varmistuksesta, joissa eri ajankohtien varmistukset on numeroitu 1–9 ja palvelinten eri varmistusosioista (engl. backup set) käytettiin pistenotaatiota siten, että pisteen vasemmalla puolella oleva numero viittaa aina yksittäiseen palvelimeen ja oikealla puolella oleva erottaa osiot toisistaan. Ensimmäiseksi taulukossa 6.3 on kuvattu kolme loogista levyosioita sisältävä hallintapalvelin (1.1–1.7) ja kaksi loogista levyosiota sisältänyt tiedostopalvelin (2.1–2.4).

Hallintapalvelin oli fyysinen Windows 2003R2 käyttöjärjestelmällä varustettu palvelin, jossa yksi loogisista levyosioista käytti kahta peilattua palvelimeen paikallisesti asennettua kiintolevyä (RAID 1) sekä kaksi muuta levyosiota levytilaa SAN-verkkoon kytketystä levyjärjestelmästä.

Taulukko 6.3. *Varmistusten nopeus kahdesta palvelimesta vanhalla varmistusjärjestelmällä.*

	1.1	1.2	1.3	1.4	1.5	1.6	1.7	2.1	2.2	2.3	2.4
1	387	2179	827	235	474	139	431	1137	1442	303	120
2	376	2182	1123	293	1729	144	454	1086	1591	330	119
3	382	2052	1240	260	1695	139	440	1167	1606	303	128
4	399	2146	943	275	1053	144	440	780	1585	303	128
5	381	2074	1415	292	1315	134	444	1119	1624	303	128
6	425	2241	1449	312	3630	139	359	616	1520	330	128
7	412	2174	733	234	1082	129	319	635	1328	330	138
8	376	2057	1290	292	3186	144	447	853	1418	303	128
9	367	1344	1160	275	1733	134	440	608	1049	279	120
keskiarvo (Mt/min)	389	2050	1131	274	1766	138	419	889	1463	309	126
keskihajonta (Mt/min)	18,9	272,5	251,5	26,9	1021,5	5,3	47,0	240,2	184,7	17,3	6,0

Taulukon sarakkeessa 1.1 on esitetty itse varmistuspalvelimen C-levyosion varmistuksen nopeus. Tälle kyseiselle paikallista kiintolevyä käyttävälle levyosiolle oli asennettu myös käyttöjärjestelmä sekä itse varmistuskopiointiin käytettävä sovellus ja ilmeisesti osittain tämän takia nopeus on jäänyt keskimäärin vain 389 megatavuun minuutissa.

Sarakkeessa 1.2 sen sijaan on sisällöltään melko samanlaisena pysyvä D-levyosio, joka sisälsi lähinnä erilaisten sovellusten asennustiedostoja. Tämän levyosion varmistaminen suoriutui aina huomattavasti C-levyosiota suuremmalla nopeudella. Sarakkeessa 1.3 on kahdeksan pienehköä tietokantaa, jotka veivät tilaa yhteensä noin gigatavun ja sijaitsivat palvelimen C-levyosiolla.

Sarakkeessa 1.4 on esitetty viiden itse varmistusjärjestelmään kuuluvan tietokannan varmistamisen nopeus. Näiden tietokantojen koko oli yhteensä alle 100 megatavua. Sarakkeen 1.5 tuloksissa on eniten hajontaa, mutta näiden syiden tarkempi tarkastelu ohitetaan, sillä tätä kirjoitettaessa vanha varmistusjärjestelmä ei ole enää käytössä. Tämä osio sisälsi palvelimen E-levyosion, jonne tallennettiin muun muassa päivittäin virtuaalipalvelinten levykuvat erillisellä varmistussovelluksella, joskaan nämä tiedostot eivät kuuluneet tässä tarkasteltavan varmistustyön varmistettaviin tiedostoihin.

Sarakkeen 1.6 tulokset ovat säännöllisesti tämän hallintapalvelimen varmistettavista osioista hitaimmat. Kyseessä on C-levyosion Volume Shadow Copy -toiminnon avulla varmistettavat tiedostot, jossa VSS-agenttina on käytetty käyttöjärjestelmän omaa VSS-tuottajaa (VSS provider, Microsoft Software Shadow Copy provider 1.0). Tämän osuuden varmistamisessa datan määrä oli säännöllisesti hyvin pieni, tyypillisesti alle 100 megatavua, joten nopeudella ei ollut suurta merkitystä koko varmistustyön kestoon.

Sarakkeessa 1.7 on lopuksi palvelimen eräänlainen tilannevedos (engl. System State Data), joka sisältää esimerkiksi järjestelmän käynnistystiedostot (engl. boot files) ja rekisterin (Microsoft TechNet). Näidenkin tiedostojen koko oli keskimäärin noin yksi gigatavu ja tiedostojen varmistamiseen käytettiin samaa edellä mainittua VSS-tuottajaa.

Taulukossa 6.3 on kuvattu myös toisen palvelimen varmistusten nopeuksia (sarakkeet 2.1–2.4). Tämä tiedostopalvelin on edellisestä poiketen virtuaalinen, mutta siitä on otettu levykuvan lisäksi palvelimeen asennetun agentin avulla tiedostotason varmuuskopioita. Käyttöjärjestelmänä oli Windows 2008R2 ja kaikki palvelimen data sijaitsee virtualisointijärjestelmän kautta hallituilla levyillä.

Sarakkeessa 2.1 on kuvattu palvelimen järjestelmälevynä toimiva C-levyosion varmistusnopeudet. Varmistettavaa dataa tällä levyosiolla oli tyypillisesti noin viisi gigatavua ja varmistusnopeus keskimäärin noin 900 megatavua minuutissa. Sarakkeessa 2.2 esitetyt varmistusnopeudet liittyvät koko palvelimen tärkeimpään dataan, palvelimen D-levyosioon, jolla sijaitsee esimerkiksi organisaation käyttäjien verkkolevyiltä käytettävät tiedostot, eli käyttäjien kotihakemistot ja eri toimipisteiden tai organisaation sisäisten yksiköiden verkkolevyt. Tältä levyosiolta siirrettiin keskimäärin noin 260 gigatavua dataa varmistusnauhoille, joten tämän osuuden kesto koko viikoittaisesta täydestä varmistuksesta oli varsin merkittävä. Luonnollisesti myös tässä varmistusosiossa saattoi käyttäjillä olla tiedosto avoinna ja käytössä varmistusten ajon ollessa käynnissä, joten VSS-toiminto oli käytössä.

Sarakkeen 2.3 osio liittyi palvelimella olleiden viiden pienehkön (noin 65 megatavua) tietokannan varmistamiseen ja sarake 2.4 sarakkeen 1.6 tavoin palvelimen C-levyaseman Shadow Copy -komponenttien varmistamiseen. Tässäkin tapauksessa nopeus jäi varsin vaatimattomasti, mutta koko järjestelmän toimintaan sillä ei ollut suurta merkitystä, koska varmistettavaa dataa oli vain noin 30 megatavua.

Kyseisen varmistustyön kaikkien palvelinten varmistusnopeudet löytyvät liitteestä A, sillä tämän diplomityön osalta ei ole olennaista käydä niitä kaikkia yksityiskohtaisesti läpi. Mainittakoon kuitenkin, että keskiarvoltaan nopein varmistustyö siirsi dataa noin 5000 megatavua minuutissa, joka alkaa olemaan jo lähellä gigabitin nopeudella toimivan verkon suorituskyvyn rajoja, mikä käytännössä viimeistään rajoittaa varmistustöiden suorituskykyä. Tarkasteltavat varmistusjärjestelmät käyttävät nopeuksien yksikkönä megatavua minuutissa tietoliikenteessä yleisesti käytetyn bittiä sekunnissa sijasta. Gigabitin verkko siirtää teoriassa 128 megatavua sekunnissa, kun käytössä on kaksikantainen järjestelmä (kilobitti = 1024 bittiä), joten minuutissa tätä gigabitin verkkoa pitkin saisi teoriassa siirrettyä 7680 megatavua. Tähän on kuitenkin mahdoton päästä, sillä esimerkiksi kohdepalvelimen levyoperaatiot ottavat oman aikansa ja koko verkon liikenne ei ole hyötykuormaa.

Näistä tuloksista voidaan kuitenkin varmuudella päätellä, että palvelinten varmistusten suorituskykyyn vaikutti vanhalla järjestelmällä suuresti kohdekoneen ominaisuudet. Nauhalle suoritettava varmuuskopiointi on myös tyypiltään sekventiaalista, eli kirjoittaminen tai lukeminen tapahtuu vain yhdessä kohdassa nauhaa kerralla. Tällöin siis useampaa varmistustyötä ei voi suorittaa käyttäen samaa nauhuria, vaan jokaiselle rinnakkaiselle työlle tarvitaan oma nauhurinsa. Sekventiaaliseen datan tallentamisen asettamiin haasteisiin palataan myös datan palauttamiseen liittyen luvussa 6.3.

Taulukossa 6.3 kuvatut tulokset olivat pieni ote käytössä olleesta viikoittaisesta D2T-varmistustyöstä, jossa kaikki data kirjoitettiin suoraan kohdekoneilta nauhalle. Vanhalla järjestelmällä oli käytössä myös D2D2T-varmistuksia päivittäin suoritettaville varmistustöille, sillä muuten toimistoaikojen ulkopuolella oleva aikaikkuna ei olisi riittänyt. Viikonloppuna suoritettavalla viikoittaisella tai kuukausittaisella varmistustyöllä oli enemmän aikaa käytettävissään, sillä varmistettavien palvelinten käyttöaste oli viikonloppuna aina paljon toimistoaikoja matalampi, joten varmistustyö sai olla myös päiväsaikaan päällä.

6.2.2 Uusi järjestelmä

Uudessa varmistusjärjestelmässä nauhojen merkitys pieneni ja levyvarmistusten merkitys kasvoi. Päivittäin suoritettavat varmistustyöt käyttävät varmistusjärjestelmän deduplikointiominaisuutta datan tallentamiseen, kun taas viikonloppuisin tehtävät viikoittaiset ja kuukausittaiset varmuuskopiot ajetaan vanhaan tapaan nauhakirjastoon, josta esimerkiksi kuukausittaiset nauhat tuodaan säännöllisesti ulos konesalista. Vaikka uuden järjestelmän yksi merkittävimmistä eroista vanhaan järjestelmään verrattuna on datan deduplikointi levyvarmuuskopioissa, on tähän alalukuun kerätty tilastoja uudella järjestelmällä ajetuista viikoittaisista nauhavarmuuskopioinneista. Vain näiden vertaaminen edellisen luvun tuloksiin on edes jollain tasolla mahdollista käyttäen numeerista dataa.

Toinen uuden järjestelmän merkittävä ominaisuus on kyky ottaa tuetuista virtuaalisista palvelimista samanaikaisesti levykuvan lisäksi tiedostotasoisia varmistuksia. Lukumäärällisesti virtuaalisia palvelimia onkin palvelininfrastruktuurissa eniten, mutta vain pienessä osassa on käytössä tämän ominaisuuden kannalta tuettu Windows-pohjainen käyttöjärjestelmä. Suurin osa sovelluspalvelimista käyttää Linuxia, josta valittu varmistusjärjestelmä ei osaa ottaa tiedostotason varmistuksia yhtäaikaisesti levykuvan kanssa. Tämän ominaisuuden ansiosta kuitenkin osasta palvelimia sai ikään kuin kaksi erilaista varmuuskopiota kerralla. Tämän vaikutusta on kuitenkin erittäin haastavaa esittää tilastoina ja vertailla vanhempaan järjestelmään.

Taulukkoon 6.4 on kerätty esimerkkinä varmistustöiden nopeuksista kahdeksasta eri palvelimesta (A–H), jotka ovat osa luvussa 6.1.2 esitettyä varmistustyötä. Loppujen seitsemän palvelimen tilastot on nähtävissä liitteessä B.

Taulukko 6.4. *Varmistusten nopeus kahdeksasta palvelimesta uudella varmistusjärjestelmällä.*

	A	B	C	D	E	F	G	H
1	1391	1780	3331	1997	2293	2330	2440	2384
2	1406	1736	3343	1980	2259	2388	2385	2529
3	2376	1602	3274	2380	2062	2386	2348	2557
4	2458	1797	3386	2634	2288	2488	2376	2665
5	2011	1520	3102	2376	2147	2341	2352	2337
6	2143	1580	3218	2564	2294	2411	2472	2621
7	2436	1862	3343	2654	2346	2465	2521	2657
8	2655	2961	2188	2364	2560	2864	2477	2429
9	2795	3233	2351	2563	2498	2837	2673	2497
10	2837	3302	2383	2633	2605	2931	2692	2562
Keskiarvo (Mt/min)	2250,8	2137,3	2991,9	2414,5	2335,2	2544,1	2473,6	2523,8
Keskihajonta (Mt/min)	491,7	685,1	456,9	237,9	164,8	223,9	117,8	106,7

Edelleen järjestelmien vertaamista toisiinsa vaikeuttaa uuden järjestelmän yksinkertaisempi tapa esittää varmistustöihin liittyvät tilastot. Tässä tapauksessa kustakin palvelimesta esitetään vain yksi nopeus, joten esimerkiksi palvelimen eri levyosioita ei ole eritelty toisistaan.

Palvelimissa A–D on käyttöjärjestelmänä 32-bittinen Windows Server 2003 ja palvelimissa E–H 64-bittinen Windows Server 2008 R2. Tämä ei kuitenkaan vaikuttaisi näkyvän tuloksissa mitenkään. Palvelimet C, F ja H sisältävät pienet SQL-tietokannat, joiden koot olivat kuitenkin lopputuloksen kannalta niin pienet, ettei niillä ole koko palvelimen varmistusnopeuden kannalta suurta merkitystä.

Palvelinten A, B ja C varmistusten keskinopeus poikkeaa eniten muista, mutta näiden keskihajonta on myös muita huomattavasti suurempi, joten nopeuksien eroista on tämänkään perusteella vaikea tehdä pitäviä johtopäätöksiä. Keskinopeuksien suuremman vaihtelun tarkempi analysointi vaatisi ainakin palvelinten sisältämien tiedostojen lukumäärien ja keskimääräisten kokojen sekä mahdollisesti varmistustöiden lokitiedostojen tarkempaa selvittämistä.

6.3 Palautusnopeus

Molemmat järjestelmät keräävät lokiin tietoja varmistustyön suorittamisesta, joten tämän ansiosta näitä kahta järjestelmää on mahdollista verrata keskenään. Datan palauttamisen nopeus on kuitenkin erittäin jossain tapauksissa erittäin riippuvainen kohdejär-

jestelmästä ja datan tyypistä, joten jälleen kerran kovin pitkälle vieviä johtopäätöksiä on tämän perusteella vaikea tehdä.

Esimerkin vuoksi tehtiin vanhalla järjestelmällä tiedostojen palautus nauhalta varmistuspalvelimen paikalliselle levylle. Tiedostoja palautuksessa oli yhteensä vain 14 kappaletta ja niistä yksi edusti datamäärän osalta enemmistöä ollen kooltaan noin 60 gigatavua. Tiedostot olivat erään suuren tietokannan vedostiedostoja (engl. dump), sekä näiden ajamiseen liittyviä skriptitiedostoja. Kyseessä oli datan palauttamisen kannalta melko ideaalisia tiedostoja, sillä suurien tiedostojen käsittely varmistusjärjestelmässä on osoittautunut selkeästi nopeammaksi kuin pienten tiedostojen. Alla sijaitsee ote palautustyön lokista.

```
Byte count          : 71 590 919 115 bytes
Rate                    : 9 103,00 MB/Min
```

Tämän kaltaisiin nopeuksiin on kuitenkin käytännössä vaikea päästä, sillä data saatetaan joutua palauttamaan lähiverkon yli ja esimerkiksi tietokantapalvelimen tapauksessa vedostiedostojen sisältö tulee vielä palauttaa erikseen käytettyyn tietokantaan.

Tätä työtä tehdessä ei ollut valitettavasti mahdollista kaikissa tapauksissa testata tuotantoa vastaavassa ympäristössä datan palauttamista. Joskin uudella järjestelmällä oli satunnaisesti tarve palauttaa esimerkiksi käyttäjien vahingossa poistamiaan sähköpostejaan. Tämän palautusnopeutta ei kuitenkaan voi mitenkään verrata äskeiseen esimerkkiin, sillä datan palauttaminen poikkeaa niin merkittävästi suurien tiedostojen palauttamisesta. Esimerkkitapauksessa käyttäjälle palautettiin deduplikointikannasta noin 4,5 megatavun kokoinen sähköposti. Alle on liitetty ote tämänkin palautustyön lokista.

```
Processed 4 534 484 bytes in 51 seconds.
Throughput rate: 5.09 MB/min
```

Tässä tapauksessa varmistusjärjestelmä on joutunut käsittelemään sähköpostipalvelimen tietokannasta varmuuskopioitua dataa, joten kyse ei ole ollut vain tiedoston palauttamisesta. Tämä ominaisuus on myös varsin usein varmistusjärjestelmissä maksullinen lisäominaisuus. Vaikka tässä tapauksessa datan palauttamisen nopeus on erittäin vaatimaton aikaisempaan esimerkkiin verrattuna, on kyseessä silti varsin nopea palautus, sillä data pystyttiin palauttamaan palvelimen paikallisella levytilalla olevasta deduplikointikannasta.

Vastaavan sähköpostiviestin palauttaminen nauhakirjastosta on huomattavasti tätä hitaampaa. Tämä johtuu siitä, että varmistuspalvelin joutuu hakemaan kyseisen sähköpostilaatikon käyttämän tietokannan tilapäisesti varmistuspalvelimen levylle ja vasta tämän jälkeen palauttamaan käyttäjän tarvitseman tiedoston. Jo yksistään oikean nauhan hakeminen nauhakirjaston nauhuriin ja nauhasta kohdan, josta tietokannan varmuuskopio alkaa, kesti kauemmin kuin koko viestin palauttaminen deduplikointikannasta kesti.

Esimerkkitapauksessa palautettiin vain noin sadan kilotavun kokoinen sähköpostiviesti käyttäen nauhakirjastossa ollutta dataa.

```
Processed 140680853009 bytes in 20 minutes and 33 seconds.
Throughput rate: 6529 MB/min

Processed 138015 bytes in 30 seconds.
Throughput rate: 0.263 MB/min
```

Tässä tapauksessa siis noin 140 gigatavun kokoisen sähköpostitietokannan lukemiseen tilapäisesti varmistuspalvelimen levylle kesti ensin 20 minuuttia 33 sekuntia. Vasta tämän jälkeen levylle siirretystä tietokannasta pääsi varmistusjärjestelmä palauttamaan käyttäjän tarvitseman sähköpostin. Yhteensä koko palautustyö vei siis yli 21 minuuttia, joten levypohjaisen varmistusjärjestelmän edut tulevat tämän esimerkin myötä varsin selkeästi esille. Tässä tapauksessa yksittäisen sähköpostiviestin palauttamisen kestossa suurin vaikutus oli kyseisen postilaatikon käyttämällä tietokannan koolla.

Loppukäyttäjälle tuo esimerkin tapauksessa kuvattu 20 minuutin ero palautusnopeudessa ei välttämättä edes juurikaan vaikuta, sillä tukipyynnön vasteaika saattaa olla vielä tätä monta kertaa suurempi. Sen sijaan on pidettävä myös mielessä se, että nauhalta datan palauttaminen vaatii myös vapaan nauhurin. Yhden nauhurin varassa oleva varmistusjärjestelmä joutuu mahdollisesti vielä odottamaan edellisen yön D2D2T-varmistustyön päättymistä, ellei tätä haluta keskeyttää. Levypohjaisesta järjestelmästä onnistuu datan palauttaminen, vaikka useampi varmistustyö kirjoittaisi samanaikaisesti uutta dataa levylle.

6.4 Deduplikoinnin vaikutus levytilan tarpeeseen

Luvussa 3.8 mainitut deduplikoinnin avulla saavutetut pakkaussuhteet eivät vaikuta täysin tuulesta tempaistuilta markkinointipuheilta. Asiakkaan käyttämässä varmistusjärjestelmässä noin 12 teratavun nettokapasiteetin omaava deduplikointilevy on osoittautunut reilun vuoden käytön aikana varsin riittäväksi, sillä käytössä siitä on noin puolet. Keskimääräiseksi deduplikointisuhteeksi järjestelmä ilmoittaa 12,1:1, joka toki vaihtelee kulloinkin varmistettavan datan mukaan. Tämä siis tarkoittaa, että varmistuspalvelimella oleva data olisi tarvinnut raakaa levytilaa yli 72 teratavua, jota ei järjestelmän hankinnan aikaan markkinoilla olevilla tuotteilla olisi saanut mahtumaan 2U:n korkuiseen räkkiasennettavaan palvelimeen ja olisi toisaalta maksanut myös huomattavasti nykyistä ratkaisua enemmän.

Eri palvelinten varmistustöiden deduplikointisuhteet vaihtelivat myös varsin paljon. Heikoin deduplikointisuhde oli säännöllisesti kahdella fyysisellä Linux-palvelimella, joista varmuuskopioitiin Oraclen tietokannan vedostiedostot inkrementaalisina varmuuskopioina. Deduplikointisuhde oli tällöin vain 1,4:1. Toisaalta korkein järjestelmän raportoima deduplikointisuhde oli 999,0:1, joka käytännössä siis tarkoittaa, että kysei-

sen palvelimen sisältämä data oli muuttunut edellisiin varmistusajoihin nähden erittäin vähän. Näiden tilastojen tarkempi tutkiminen olisi ollut varsin mielenkiintoista, mutta järjestelmän omasta raportointityökalusta ei ollut mahdollista viedä dataa taulukkomuodossa taulukkolaskentaohjelmalla suoritettavaa tilastointia varten.

6.5 Deduplikoinnin vaikutus varmistuspalvelimen kuormitukseen

Varmistusjärjestelmän hankinnassa oli varauduttu deduplikointitekniikan aiheuttamiin korkeampiin suorituskykyvaatimuksiin varustamalla palvelin kahdella suorittimella. Muistia varmistusjärjestelmään oli myös asennettu reilusti valmistajan suositusta enemmän, sillä 12 teratavun deduplikointikannalle tulee valmistajan suositusten mukaisesti varata 18 gigatavua muistia. Käytännössä deduplikointikannan kooksi vakiintui noin 6 teratavua, joten järjestelmässä on tämän puolesta selkeästi kasvun varaa. Käytännön havainnoissa ja käyttöjärjestelmän valvontatyökaluilla suoritettujen mittauksien perusteella voidaan todeta, että ainakin suorittimen ja muistin resurssit riittävät reilulla marginaalilla deduplikoinnin käyttöön.

6.5.1 Suorittimen ja muistin käyttöaste

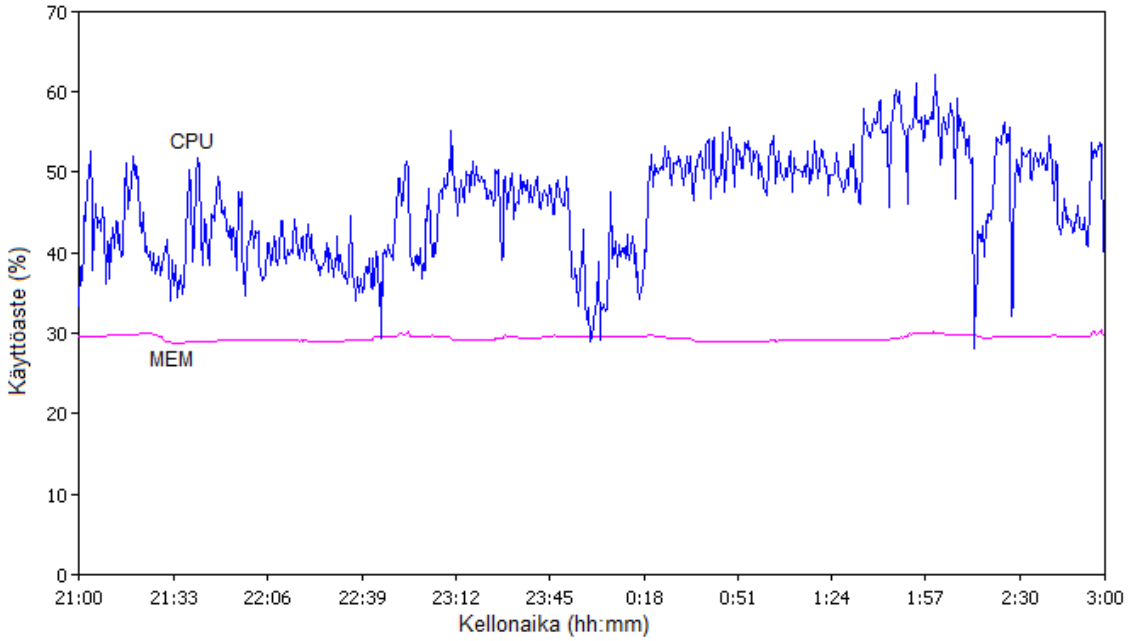
Suorittimen ja muistin käyttöastetta mitattiin neljänä perättäisenä päivänä siten, että varmistusjärjestelmässä oli vähintään kolme rinnakkaista deduplikoivaa varmistustyötä yhtäaikaisesti käynnissä. Dataa kerättiin Perfmon.exe-työkalulla yhteensä kuusi tuntia klo 21.00 alkaen ja tämän aikavälin keskiarvot on kerätty taulukkoon 6.5.

Taulukko 6.5. *Varmistuspalvelimen suorittimen ja muistin käyttöaste.*

	Suorittimen käyttöaste (%)	**Muistin käyttöaste (%)**
1	46,16	29,41
2	47,40	33,13
3	47,06	36,96
4	47,66	40,36
Keskiarvo	47,07	34,97

Näiden mittaustulosten perusteella järjestelmän suorituskyky prosessoritehon ja muistin osalta on täysin riittävä. Muistin lisääminen ja prosessorien vaihtaminen nopeammaksi on myös varsin helppoa ja kustannuksiltaan edullista, mikäli nämä tulevaisuudessa osoittautuisivat pullonkaulaksi. Todennäköisesti kuitenkin kyseisen varmistusjärjestelmän noin neljän vuoden elinkaaren aikana tähän ei tule olemaan tarvetta. Kuvassa 6.1 on esitetty edellä mainitun taulukon ensimmäisestä mittausajanjakson tulokset kuvaajana, josta käy tarkemmin ilmi esimerkiksi prosessorin kuormituksen käyttäytyminen. Y-

akselilla kuvaajassa on prosessorin (CPU) ja muistin (MEM) käyttöaste prosentteina sekä x-akselilla kellonaika. Taulukon 6.5 kolmen muun mittauksen kuvaaja löytyy liitteestä C.



Kuva 6.1. *Varmistuspalvelimen prosessorin (sinisellä) ja muistin (punaisella) käyttöaste prosentteina.*

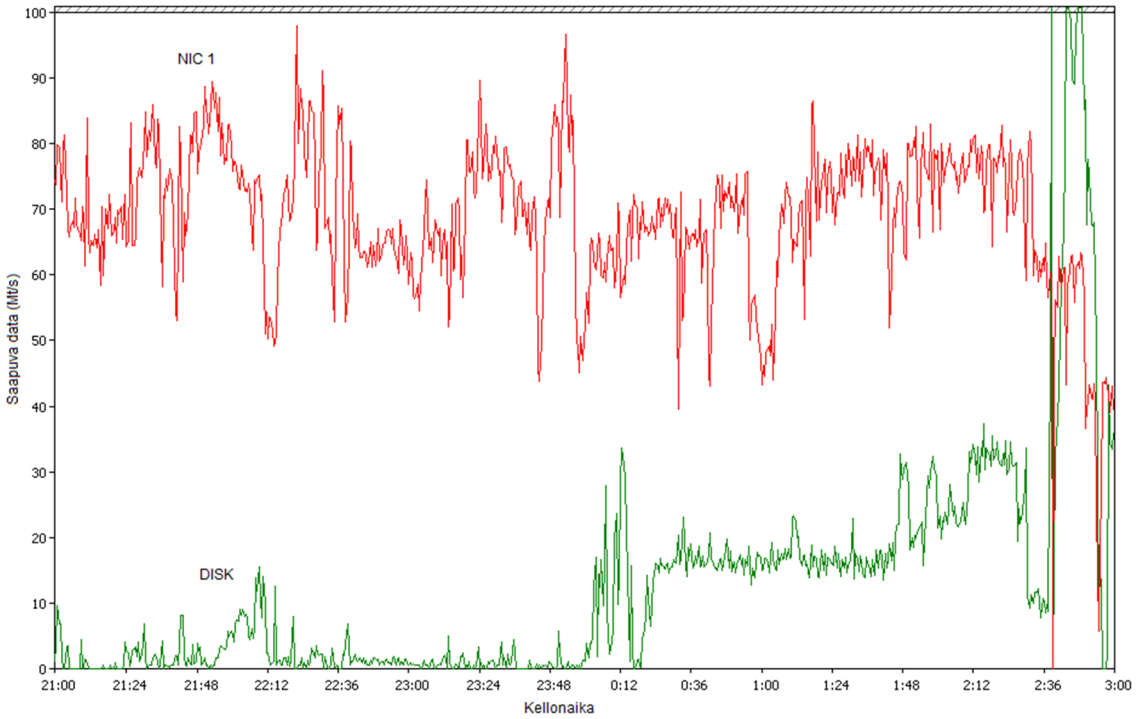
Muistia järjestelmä on allokoinut käyttöönsä melko tasaisesti koko mittausjakson ajan, eikä prosessorikuormassakaan esiinny valtavan suuria piikkejä. Kenties huomioin arvoista on se, että säännöllisesti keskiyön aikaan on järjestelmän prosessorin käyttöasteessa selkeä notkahdus. Tämän perusteella voidaan tulkita, että tuona ajankohtana kohdalle osuu varmistettava palvelin, jonka datan deduplikointi on prosessorille poikkeuksellisen kevyttä. Usean samaan aikaan ajetun varmistustyön johdosta tämän aiheuttajaa on kuitenkin vaikea selvittää.

6.5.2 Verkon ja levyjen suorituskyky

On kuitenkin muistettava, että palvelimen prosessori ja muisti eivät ole ainoita resursseja, joita datan deduplikointi palvelimelta tarvitsee. Esimerkiksi käytettävän levyohjaimen ja levyjen tulee olla riittävän suorituskykyisiä tähän tarkoitukseen. Taulukkoon 6.6 on koottu yhteenveto vastaavalla tavalla palvelimen kummankin verkkoliitännän (NIC 1 ja NIC 2) saapuva liikenne, deduplikointikannan sisältämän loogisen levyaseman kirjoitus- ja lukuoperaatioiden lukumäärä (IOPS, Input/Output Operations Per Second) sekä kirjoitus- ja lukunopeus käyttäen samaa työkalua kuin luvussa 6.5.1.

Taulukko 6.6. *Varmistuspalvelimen verkkoliitäntöjen saapuva liikenne, levyn kirjoitus- ja lukuoperaatioiden lukumäärä sekä kirjoitus- ja lukunopeus.*

	NIC 1 in (Mt/s)	NIC 2 in (Mt/s)	IOPS (kpl/s)	Kirjoitusnopeus (Mt/s)	Lukunopeus (Mt/s)
1	68,76	0,002	364,26	12,61	10,00
2	69,23	0,002	196,18	6,10	6,48
3	68,85	0,002	189,22	5,92	6,47
4	69,01	0,002	187,56	6,11	6,33
Keskiarvo	68,96	0,002	234,31	7,69	7,32



Kuva 6.2. *Varmistuspalvelimelle saapuvan verkkoliikenteen nopeus (punaisella) ja kirjoitusnopeus levylle (vihreänä), yksikkönä Mt/s.*

Taulukon 6.6 tuloksissa, ja erityisesti kuvasta 6.2, on havaittavissa, että levylle kirjoittamisen nopeus oli selkeästi alhaisempi kuin mitä verkkoportin kautta sisään tuli dataa. Tämä on toisaalta loogista, sillä varmistusjärjestelmän deduplikointiprosessi ei kirjoita redundanttia dataa tietokantaan tai edes väliaikaisesti levylle, vaan käsittelee datan palvelimen RAM-muistissa. Liitteessä D löytyy taulukon 6.6 tuloksista muodostetut kuvaajat, joista ensimmäinen on esitetty kuvassa 6.2.

Edellisen lisäksi viikonlopun jälkeen ajettu varmistustyö (Taulukko 6.6, rivi 1) kirjoitti ja luki levyltä selvästi enemmän dataa kuin tätä seuraavat päivittäiset varmistustyöt. Tämä selittyy rivien 2–4 paremmalla datan deduplikointiasteella, sillä maanantaiyönä ajettua varmistustyötä edellinen deduplikointivarmistustyö on suoritettu torstain ja per-

jantain välisenä yönä. Perättäisinä päivinä palvelinten sisällössä tapahtunut muutos on pienempi kuin torstain ja maanantain välisenä ajanjaksona.

Mielenkiintoinen yksityiskohta on myös havaittavissa kuvan 6.2 kuvaajasta aivan havaintojakson lopusta. Levylle kirjoitetun datan nopeus nousi selkeästi suuremmaksi kuin mitä verkkoliitännän kautta sitä saapui palvelimelle. Vastaava tilanne ei toistunut seuraavina päivinä (liite D), vaan muissa perättäisten päivien kuvaajissa oli havaittavissa samanlaisena toistuvia piirteitä.

Varmistuspalvelimen levyohjaimen ja levyjen suorituskykyä testattiin alun perin Intelin kehittämällä avoimeen lähdekoodiin perustuvalla Iometer-nimisellä sovelluksella (Iometer). Työkalulla luotiin hyvin pelkistetyillä asetuksilla keinotekoista kirjoitus- ja lukukuormaa palvelimen loogiselle E-levyosiolle, joka käyttää samoja 23 levyä, kuin deduplikointikannan käyttämä D-levyosio. Tarkemmat tiedot käytetyistä asetuksista on kerätty liitteeseen E. Tällä tavoin järjestelmää kuormittamalla saatiin Perfmon-sovelluksella mitattua kirjoitus- ja lukunopeudeksi noin 300 megabittiä sekunnissa, eli yhteensä siis noin 600 megabittiä sekunnissa.

Levylle suoritettuja kirjoitusoperaatioita (IOPS, Input/Output Per Second) oli keskimäärin noin 2050 sekunnissa. Varmistusjärjestelmän deduplikointiprosessi oli koko ajan käynnissä, mutta yhtään varmistustyötä ei ollut käynnissä. Deduplikointiprosessi kuormittaa levyjä kuitenkin jatkuvasti, suorittaen enimmäkseen lukuoperaatioita, jotka liittyvät ilmeisesti kannan ylläpitoon. Tämän prosessin toiminnasta ei löytynyt tarkempaa dokumentaatiota. Tämän vaikutus kuormitustestin tuloksiin osoittautui kuitenkin häviävän pieneksi.

Taulukon 6.6 tuloksista voidaan todeta, että deduplikointia käyttävät varmistustyöt kirjoittivat ja lukivat keskimäärin yli puolet hitaammin, kuin mitä Iometer-sovelluksella palvelimelle pystyttiin luomaan keinotekoista kuormaa. Vielä suurempi ero oli yksittäisten kirjoitus- ja lukuoperaatioiden määrällä, sillä ero testikuormitukseen oli noin kymmenkertainen. Tämän perusteella voidaan tulkita, että myöskään palvelimen levyohjaimen ja levyjen nopeudet eivät muodostuneet järjestelmän pullonkaulaksi.

Toisaalta tuloksista käy myös selkeästi ilmi, että palvelimeen kytketyistä verkkoliitännöistä toisen läpi ei saavu mainittavaa määrää dataa. Verkkoyhteys näkyy käyttöjärjestelmälle verkkoliitäntänä, jonka nimellinen nopeus on kaksi gigabittiä sekunnissa, mutta näiden kahden verkkoliitännän yhdistettyä kapasiteettia järjestelmä ei kuitenkaan pysty hyödyntämään täysin.

Palvelimen valmistaja käyttää tästä usean verkkoliitännän yhtäaikaisesta käyttämisestä termiä *NIC teaming*, joita on kuusi toimintatavaltaan erilaista. Käytössä oleva palvelin on asetettu valitsemaan automaattisesti sopivin käytettävissä oleva vaihtoehto. Tässä tapauksessa käytössä on Transmit Load Balancing with Fault Tolerance (TLB), jossa

nimensä mukaisesti kuormaa jaetaan vain dataa lähetettäessä, mutta ominaisuus lisää silti vikasietoisuuden verkkoyhteyteen. (Hewlett-Packard 2008, s. 13–14)

Varmistuspalvelimen kannalta olisi kuitenkin useimmissa tapauksissa tärkeämpää kyetä vastaanottamaan dataa useamman verkkoliitännän kautta, sillä tällä olisi suora vaikutus varmistustöiden nopeuteen. Tähän tarkoitukseen sopisi siten paremmin IEEE:n standardiin 802.3ad perustuva dynaaminen ja vikasietoinen kuormanjako, jota palvelimen valmistaja käyttää nimitystä 802.3ad Dynamic with Fault Tolerance. Tämä mahdollistaisi kuorman jakamisen eri verkkoliitäntöjen kesken sekä lähetettäessä että vastaanotettaessa dataa. Ominaisuus vaatii kuitenkin tuen käytetyltä kytkimeltä. (Hewlett-Packard 2008, s. 14)

Giljae et al. ovat Iperf-sovelluksella suorittamissaan testeissä havainneet, että gigabitin Ethernet-verkon kehyksen suurin sallittu koko (MTU, Maximum Transmission Unit) vaikuttaa merkittävästi verkon läpimenonopeuteen (engl. throughput), kun päästä−päähän-viive (RTT, Round Trip Time) on suuri. Käytettäessä oletusarvoa 1500 tavua kehyksen suurimpana kokona, saavutettiin testeissä verkkokortista riippuen noin 500–600 megabitin sekuntinopeus, kun RTT:n arvo oli 228 millisekuntia. Kun kehyksen maksimikooksi asetettiin 9000 tavua, pääsi kaikki viisi testissä käytettyä verkkokorttia noin 800 megabitin nopeuteen. (Giljae et al. 2008) Tämä tulee ottaa siis erityisesti huomioon silloin, kun varmistuspalvelimen ja kohdepalvelimen välinen viive on suuri ja siirrettävää dataa on paljon.

Asiakkaan käytössä olevassa virtuaaliympäristössä isäntäpalvelinten sekä itse varmistuspalvelimen verkkoliitännän MTU-arvoksi on määritetty 1500 tavua, mutta RTT on vain 0,8 millisekuntia varmistuspalvelimen ja ESXi-isäntäpalvelimen välillä. Internet Engineering Task Forcen (IETF) julkaisemassa muistiossa on gigabitin verkolle laskettu TCP-protokollan teoreettiseksi läpimenonopeudeksi yhden millisekunnin RTT:llä 949,2 megabittiä sekunnissa, kun TCP:n vastaanottoikkuna tai lähetyspuskuri eivät rajoita nopeutta (RFC 6349).

Sen sijaan Mathis et al. (1997) on johtanut suurimmalle mahdolliselle kaistanleveydelle (BW, engl. bandwith) kaavan

$$BW_{max} = \frac{MSS}{RTT}\frac{C}{\sqrt{p}}, \tag{6.1}$$

jossa *MSS* (Maximum Segment Size) on TCP-segmentin suurin sallittu koko, *RTT* viive, *C* käytetystä vastaanottokuittausmenetelmästä riippuva vakio ja *p* on pakettihävikki. Vakiona *C* voidaan yksinkertaistuksen vuoksi käyttää arvoa 1.

Mikäli Ethernet-kehyksen MTU on 1500 tavua, niin TCP-segmentin suurin koko on tuolloin 1460 tavua. Oletetaan pakettihäviöksi 10^{-4} eli 0,01 %, joka on testien mukaan

mahdollista saavuttaa käytetyssä konesaliympäristössä. Tällöin suurimmaksi saavutettavissa olevaksi kaistanleveydeksi saadaan kaavan 6.1 mukaisesti

$$BW_{max} = \frac{1460\ B*8\ bit/B}{0{,}0008\ s}\ \frac{1}{\sqrt{10^{-4}}} = 1460\ \frac{Mbit}{s}. \quad (6.2)$$

Tämän perusteella voidaan todeta, että varmistusjärjestelmän käyttävä verkkoliikenne ei saavuta missään vaiheessa suurinta mahdollista verkon teoreettista läpimenonopeutta. Järjestelmä siirtää dataa kuitenkin tuotantokäytössä olevasta virtuaalipalvelinympäristöstä, joka ei todennäköisesti vastaa optimaalisia laboratorio-olosuhteita. Käyttöjärjestelmän ylläpitotyökaluja ja varmistusjärjestelmän hallintasovellusta tarkastelemalla on myös todettavissa, että varmistustyöt eivät siirrä dataa koko ajan täydellä nopeudella, vaan aikaa kuluu esimerkiksi virtuaalipalvelimista tilannekuvan (engl. snapshot) ottamiseen ja tämän poistamiseen sekä muihin valmisteleviin toimenpiteisiin.

Koska konesaleissa olevat kytkimet ovat jo elinkaarensa lopussa, olisi luonnollista näiden uusimisen yhteydessä ottaa käyttöön seuraavan sukupolven 10 gigabitin nopeuteen kykenevät kytkimet. Näin ollen nykyiseen konfiguraation muuttamiseen ja muutosten toiminnan testaamiseen ei kannata uhrata resursseja, sillä odotettavissa oleva läpimenonopeuden kasvu on kohtuullisen vaatimatonta verrattuna siirtymiseen 10 gigabitin nopeutta tukevaan verkkoinfrastruktuuriin.

Wu-chun et al. tutkivat varsin kattavasti 10 gigabitin nopeudella toimivan Ethernet-verkon läpimenonopeutta jo vuonna 2003. Tuolloin käytössä oli kolme erilaista verkkotopologiaa verkkoyhteyden pullonkaulan selvittämistä varten. Näissä testeissä saavutettiin 4,11 gigabitin sekuntinopeus käyttäen MTU:n arvona 8160 tavua. Suoritetuissa testeissä kävi kuitenkin ilmi, että pullonkaulana ei ollut itse verkko tai verkkokortti, vaan palvelimen sisäinen väylä. (Wu-chun et al. 2003) Palvelimissa käytetyt tekniikat ovat tästä kehittyneet varmasti merkittävästi, mutta tämä osoittaa selkeästi sen, että verkkoliitännän teoreettiseen nopeuden saavuttaminen ei ole täysin itsestään selvää. Konesalissa siirtyminen 10 gigabitin nopeudella toimivien lähiverkkojen käyttöön toisi kuitenkin hyvin suurella todennäköisyydellä merkittäviä parannuksia varmistusjärjestelmän nopeuteen.

Giljae et al. testasivat myös TCP-protokollan vastaanottoikkunan koon (RCV WND, Receiver Window) vaikutusta läpimenonopeuteen gigabitin verkkoa käyttämällä. Tuloksista ilmeni, että ikkunan koolla on merkittävä vaikutus läpimenonopeuteen, sillä 65 kilotavun kokoisella ikkunalla ei päästy millään verkkokortilla edes yli 50 megabitin nopeuteen. Vastaavasti kasvattamalla ikkunan kokoa 32 megatavuun, päästiin jo lähemmäs 800 megabitin nopeuteen. Käytetyssä testiympäristössä päästä−päähän-viive oli 228 millisekuntia ja MTU:n arvona 9000 tavua. (Giljae et al. 2008)

Edellä mainittu testitulosta ei voi kuitenkaan suoraan verrata tässä diplomityössä tarkasteltuun ympäristöön, sillä esimerkiksi varmistusjärjestelmän ja virtuaaliympäristön isän-

täkoneen välinen RTT on vain noin 0,8 millisekuntia. Lisäksi esimerkiksi Microsoftin käyttöjärjestelmäversiosta Windows Server 2008 lähtien on käytössä ollut vastaanottoikkunan automaattinen säätö (engl. Receive Window Auto-Tuning), jossa hyödynnetään esimerkiksi RFC 1323:ssa esitettyä vastaanottoikkunan skaalausta, eli mahdollistetaan 65 kilotavua suurempien vastaanottoikkunoiden käyttäminen (Davies 2007). Koska läpimenonopeus on vastaanottoikkunan ja RTT:n osamäärä, niin yhden gigabitin nopeus voidaan teoriassa saavuttaa 0,8 millisekunnin RTT:llä käyttäen 100 kilotavun kokoista vastaanottoikkunaa.

$$Throughput = \frac{RCV\ WND}{RTT} \tag{6.3}$$

$$RCV\ WND = Throughput \times RTT = 10^9\ \frac{bit}{s} \times 0{,}0008\ s\ \div 8\ \frac{bit}{bytes} = 100\ kB$$

Varmistusjärjestelmän ja koko muun palvelinympäristön kannalta verkon optimaalisten asetusten hakeminen olisi mielenkiintoinen tutkimuskohteena ja varmasti riittävän laaja kokonaisen diplomityön aiheeksi. Tämän asian tarkempi tarkastelu kuitenkin ohitetaan, sillä se menee hieman ohi tämän diplomityön aiheen rajauksen.

7. YHTEENVETO

Tämän diplomityön loppuun on koottu uudella järjestelmällä saavutetut hyödyt, joiden yksityiskohtia on tarkemmin esitelty esimerkiksi luvuissa 5 ja 6. Tämän lisäksi loppuun on koottu järjestelmän käyttöönoton ja noin puolentoista vuoden käytön aikana hyväksi havaitut käytänteet.

7.1 Saavutetut hyödyt

Uudella varmistusjärjestelmällä kyetään sekä varmistamaan, että palauttamaan dataa nopeammin ja monipuolisemmin. Erityisesti deduplikoiva levypohjainen varmistusjärjestelmä nopeutti pienten yksittäisten tiedostojen ja erityisesti sähköpostiviestien palauttamista. Nauhakirjaston nauhurin vapautumista ei tarvinnut odottaa ja myös itse tiedoston palauttaminen oli nopeammin suoritettu deduplikointikannasta.

Mediana nauhakirjaston nauhat ja deduplikointikannan levyt myös kahdentavat tavallaan varmistusjärjestelmän, vaikka fyysisesti samassa sijainnissa ovatkin. Esimerkiksi deduplikointikannan korruptoiduttua pystyttiin kriittisimmistä järjestelmistä ottamaan varmuuskopiot ilman deduplikointia varmistusjärjestelmän erilliselle levyosiolle ja nauhakirjastoon.

Uusi järjestelmä toi mukanaan tuen myös uusille käyttöjärjestelmille ja varmuuskopioiden ottamiselle levykuvana suoraan virtuaaliympäristöstä. Aikaisemmin levykuvien varmistamiseen ja tiedostotasolla otettaviin varmuuskopioihin oli käytetty kahta eri sovellusta. Myös Windows-palvelimista saatava samanaikainen levykuva ja tiedostotasoinen varmuuskopio helpottavat varmistusjärjestelmän ylläpitoa ja toivat monipuolisemmat mahdollisuudet tietojen palauttamiseen. Ominaisuudesta olisi hyötyä myös Linux-palvelinten osalta, varsinkin kun liiketoiminnan kannalta kriittisimmät palvelut toimivat juuri Linux-ympäristössä.

Varmistusjärjestelmän päivitys todettiin myös välttämättömäksi, jotta liiketoiminnan jatkuvuus pystyttäisiin takaamaan muuttuvassa tietojärjestelmäinfrastruktuurissa ja varmistusjärjestelmän kapasiteetti pysyisi varmistettavan datamäärän kasvun mukana.

7.2 Havaitut parhaat käytänteet

Varmistusjärjestelmän käyttöönotossa tulisi hyödyntää muiden käyttäjien ja valmistajien kertomia parhaita käytänteitä, sillä varmistusjärjestelmän testaaminen tuotantoympäristössä on hankalaa. Täydellistä tuotantoympäristön replikaa, jossa uutta käyttöön

otettavaa varmistusjärjestelmää voisi tuotantoympäristöstä erillään testata, ei monessakaan organisaatiossa ole varaa ylläpitää. Käyttöönottoa seuraavien päivien ajaksi tulisi varata myös riittävästi resursseja järjestelmän toiminnan hienosäätämiseen ja monitoroida tarkkaan varmistustöiden etenemistä, jotta mahdollisiin ongelmatapauksiin pystytään puuttumaan välittömästi.

Varmistettaville palvelimille asennettavat agenttisovellukset vaativat toimiakseen yhteensopivan käyttöjärjestelmän, joten käyttöön otettava varmistusjärjestelmä tulee valita tämän mukaisesti. Tyypillisesti varmistusjärjestelmää uudempaan versioon päivitettäessä putoaa myös tuki vanhemmilta käyttöjärjestelmiltä, joten tämän vuoksi palvelinympäristö tulisi pitää mahdollisimman ajantasaisena. Virtuaaliympäristöjen varmuuskopioinnissa tulee lisäksi ottaa huomioon itse virtuaaliympäristön yhteensopivuus varmistusjärjestelmän kanssa. Erityistä huomioita yhteensopivuuteen tulee kiinnittää sähköpostipalvelinten, AD-palvelinten ja tietokantapalvelinten kanssa, mikäli varmistusjärjestelmällä halutaan pystyä palauttamaan esimerkiksi yksittäisiä sähköpostiviestejä. Näihin palvelimiin varmistusjärjestelmä integroituu erittäin tiiviisti ja datan palauttamisessa ei ole välttämättä kysymys vain yksittäisistä tiedostoista.

Tallennuskapasiteetti tulee varmistusjärjestelmässä resursoida riittäväksi tai hankintavaiheessa varautua kapasiteetin myöhempään lisäämiseen. Esimerkiksi liian pienen nauhakirjaston hankinta saattaa tarpeiden kasvaessa tulla kalliiksi, sillä pieneksi jääneen rinnalle saatetaan joutua ostamaan toinen tai tilalle kokonaan uusi. Järjestelmällä tulee olla myös riittävän nopea LAN- tai SAN-verkkoyhteys, jotta varmistustyöt saadaan suoritettua annetussa aikaikkunassa.

Suorituskykyyn vaikuttaa olennaisesti myös varmistettavan datan ominaisuudet, sillä esimerkiksi paljon pieniä tiedostoja sisältävän palvelimen tiedostotasoinen varmuuskopiointi kestää huomattavasti pidempään kuin suuria tiedostoja sisältävän palvelimen. Palvelinten sisältämän datan sisältöön tulisi kiinnittää myös huomioita, jotta arkistodatasta ei oteta päivittäisiä varmuuskopioita.

LÄHTEET

Alhazmi, O.H., Malaiya, Y.K. (2012). Assessing Disaster Recovery Alternatives: On-site, Colocation or Cloud. IEEE 23rd International Symposium on Software Reliability Engineering Workshops. pp. 19–20.

Bhagwat, D., Eshghi, K., Long, D.D.E., Lillibridge, M. (2009). Extreme Binning: Scalable, Parallel Deduplication for Chunk-based File Backup. IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems. pp. 1–9.

Cleveland, F.M. (2008). Cyber security issues for Advanced Metering Infrasttructure (AMI). IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century. pp. 1–5.

Dapeng, J., Chuanyi, L., Dongsheng, W., Hong, L., Zhizhong, T. (2009). Performance Comparison of IP-Networked Storage. Tsinghua Science and Technology. Vol.14(1). pp. 29–40.

Davies, J. (2007). The Cable Guy - TCP Receive Window Auto-Tuning. Microsoft TechNet Magazine. Saatavissa (viitattu 5.10.2014): http://technet.microsoft.com/fi-fi/magazine/2007.01.cableguy(en-us).aspx.

ENISA - European Network and Information Security Agency (2012). Introduction to Return on Security Investment - Helping CERTs assessing the cost of (lack of) security. 13 p. Saatavissa (viitattu 13.10.2014): http://www.enisa.europa.eu/activities/cert/other-work/introduction-to-return-on-security-investment.

Faritha Banu, A., Chandrasekar, C. (2012). A Survey On Deduplication Methods. International Journal of Computer Trends and Technology. Vol.3(3), pp. 364–368. Saatavissa: http://www.ijcttjournal.org/Volume3/issue-3/IJCTT-V3I3P108.pdf.

FIPS 200 - Federal Information Processing Standards. (2006). Minimum Security Requirements for Federal Information and Information Systems. National Institute of Standards and Technology. Saatavissa (viitattu 9.10.2014): http://csrc.nist.gov/publications/fips/fips200/FIPS-200-final-march.pdf.

Giljae, L., Yoonjoo, K., Kwangjong, C., Woojin, S., Jaiseung, K. (2008). Performance Evaluation of Gigabit Ethernet Interfaces. IEEE 10th International Conference on Advanced Communication Technology. Vol.1, pp. 114–117.

Helin, O. (2012). A Study on Virtualization and Energy Efficiency Using Linux. Master of Science Thesis. Tampere University of Technology. 56 p.

Hewlett-Packard. (2008). HP ProLiant Network Adapter Teaming. Saatavissa (viitattu 5.11.2014): ftp://ftp.compaq.com/pub/products/servers/networking/proliant-teaming-whitepaper-march%2008.pdf.

Hokkanen, H. (2007). Design of a secure free software backup system for Linux servers. Master of Science Thesis. Tampere University of Technology. 61 p.

Huffman, A.D. (1952). A Method for the Construction of Minimum-Redundancy Codes. Proceedings of the IRE. Vol.40(9), pp. 1098–1101.

IBM. (2012). Introduction to Storage Area Networking and System Networking. ISBN 0738437131. p. 338 Saatavissa (viitattu 21.4.2013): https://www.redbooks.ibm.com/redbooks/pdfs/sg245470.pdf.

IEEE Standard 802.3-1998. (1998). Part 3: Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications. p. 1222. Saatavissa (viitattu 21.4.2013): http://standards.ieee.org/getieee802/download/802.3-2008_section4.pdf.

INCITS - InterNational Committee on Information Technology Standards. Introduction to T10. Saatavissa (viitattu 21.4.2013): http://www.t10.org/intro.htm.

Iometer. Introduction. Saatavissa (viitattu 3.10.2014): http://www.iometer.org/.

LTO. What is LTO Technology?. Hewlett-Packard, IBM and Quantum. Saatavissa (viitattu 24.9.2014): http://www.lto.org/technology/what-is-lto-technology/.

LVM - Liikenne- ja viestintäministeriö. (2013). Big data avaa uusia mahdollisuuksia. Saatavissa (viitattu 12.10.2014): http://www.lvm.fi/uutinen/4156868/big-data-avaa-uusia-mahdollisuuksia.

Mathis, M., Semke, J., Mahdavi, J. (1997). The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm. ACM SIGCOMM Computer Communication Review. Vol. 27, pp. 67–82.

Microsoft TechNet. Backing Up System State Data. Saatavissa (viitattu 28.7.2014): http://technet.microsoft.com/en-us/library/cc938537.aspx.

SP 800-34 - NIST Special Publication 800-34 Rev. 1. (2010). Contingency Planning Guide for Federal Information Systems. National Institute of Standards and Technology. Saatavissa (viitattu: 9.10.2014): http://csrc.nist.gov/publications/nistpubs/800-34-rev1/sp800-34-rev1_errata-Nov11-2010.pdf.

Oracle® Database Security Guide. Oracle. Saatavissa (viitattu 13.3.2014): http://docs.oracle.com/cd/E11882_01/network.112/e36292/auditing.htm#DBSEG0622.

Paulsen, K. (2011). 12 - Archives, Backups, and Linear Tape, In Moving Media Storage Technologies. Focal Press, Boston, USA. pp. 327–377.

RFC 6349. (2011). Framework for TCP Throughput Testing. Internet Engineering Task Force. Saatavissa (viitattu: 2.11.2014): http://tools.ietf.org/html/rfc6349.

Rosenblum, M., Garfinkel, T. (2005). Virtual machine monitors: current technology and future trends. IEEE Computer Society. Vol.38(5). pp. 39–47.

SNIA. About the SNIA - The Storage Networking Industry Association. Your Connection is Here. Saatavissa (viitattu 31.3.2013): http://www.snia.org/about.

Symantec. (2012). Symantec Backup Exec 2012 Administrator's Guide. Symantec Corporation. Saatavissa (viitattu 10.8.2014): http://www.symantec.com/business/support/index?page=content&id=doc5211.

Symantec Community. (2012). Backup multiple servers in one job to one tape. Backup exec 2012. Saatavissa (viitattu 10.8.2014): http://www.symantec.com/connect/ ideas/backup-multiple-servers-one-job-one-tape-backup-exec-2012.

Symantec White Paper. (2014). Symantec Backup Exec™ 2014 Feature Comparison Matrix. Saatavissa (viitattu 10.8.2014): http://www.symantec.com/content/en/us/enterprise/white_papers/b-backup-exec-2014-feature-comparison-matrix.pdf.

Tandberg. (2013). Disk-to-Disk-to-Tape (D2D2T). Saatavissa (viitattu 28.8.2014): http://tandbergdata.com/default/assets/File/white_papers/WP-D2D2T_EN_web.pdf.

VAHTI 1/2002. (2002). Tietoteknisten laitetilojen turvallisuussuositus. Valtiovarainministeriö. Saatavissa (viitattu 8.10.2014): https://www.vm.fi/vm/fi/04_julkaisut_ja_asiakirjat/01_julkaisut/05_valtionhallinnon_tietoturvallisuus/20020101Tietot/turvallisuussuositus.pdf.

VAHTI 2/2013. (2013). Toimitilojen tietoturvaohje. Valtiovarainministeriö. Saatavissa (viitattu: 8.10.2014): http://www.vm.fi/vm/fi/04_julkaisut_ja_asiakirjat/01_julkaisut/05_valtionhallinnon_tietoturvallisuus/20130530Toimit/Toimitilojen_tietoturvaohje_VAHTI_2_2013_netti.pdf.

Valtiovarainministeriö. Julkisen hallinnon ICT - Tietoturvallisuus. Saatavissa (viitattu 8.10.2014): http://www.vm.fi/vm/fi/16_ict_toiminta/009_Tietoturvallisuus/index.jsp.

VMware. VMware vMotion: Virtual Machine Live Migration. VMware, Inc. Saatavissa (viitattu 22.10.2014): http://www.vmware.com/products/vsphere/features/vmotion.html.

Wu-chun, F., Hurwitz, J.G., Newman, H., Ravot, S., Les Cottrell, R., Martin, O., Coccetti, F., Cheng Jin, Xiaoliang Wei, Low, S. (2003). Optimizing 10-Gigabit Ethernet for

Networks of Workstations, Clusters, and Grids: A Case Study. ACM/IEEE Conference on Supercomputing. p. 50.

LIITE A: VANHAN JÄRJESTELMÄN VIIKOTTAISEN VARMISTUKSEN TILASTO

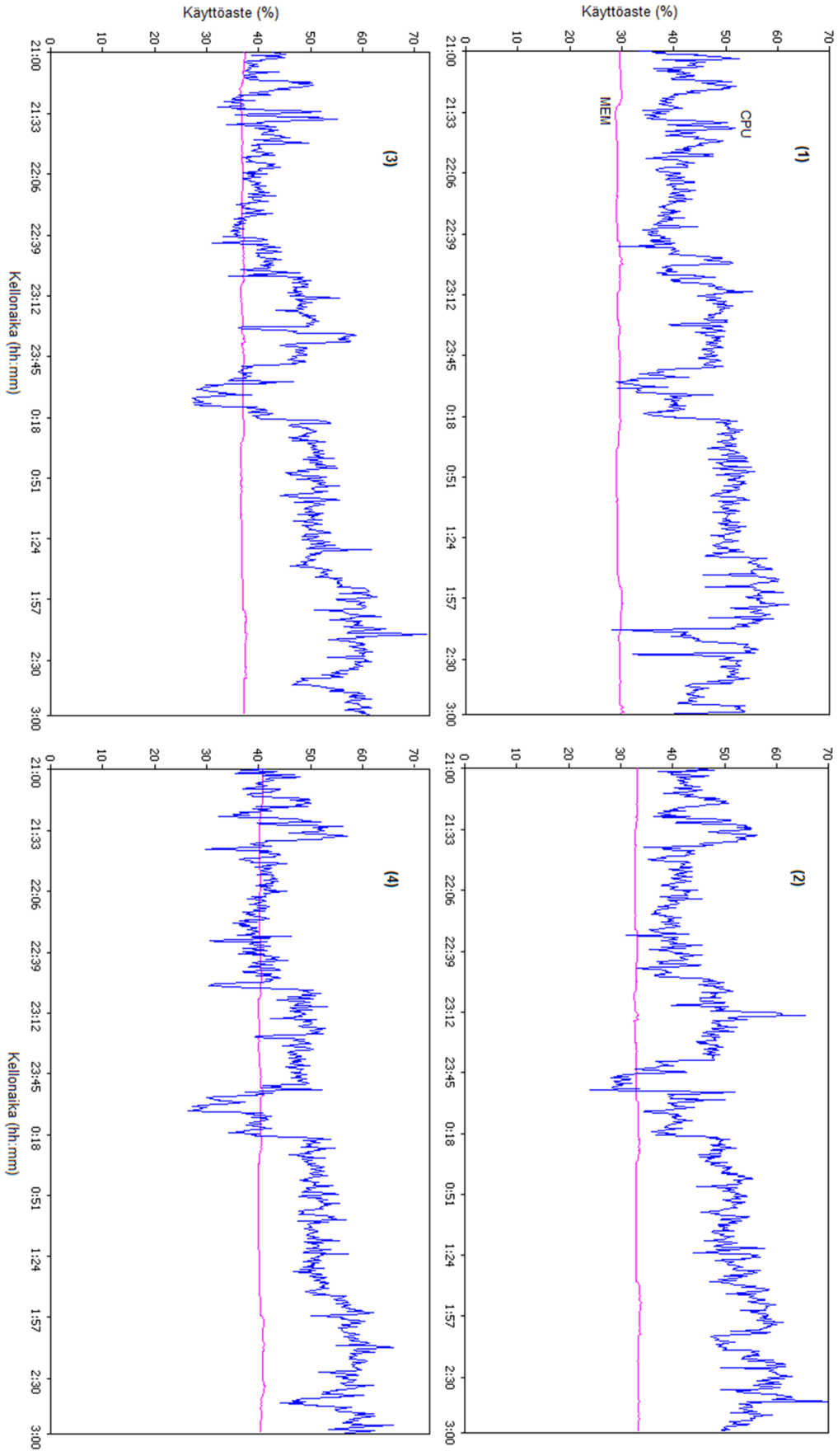
	1.1	1.2	1.3	1.4	1.5	1.6	1.7	2.1	2.2	2.3	2.4	3.1	4.1	4.2	4.3	4.4	5.1	5.2	5.3	6.1	6.2	6.3	6.4	6.5	6.6	7.1	7.2	7.3
1	387	2179	827	235	474	139	431	1137	1442	303	120	432	896	3394	214	585	1379	203	676	2349	4933	3397	1351	322	555	1144	212	641
2	376	2182	1123	293	1729	144	454	1086	1591	330	119	449	885	3562	214	591	1443	213	691	2688	5351	4262	1583	338	629	1304	200	699
3	382	2052	1240	260	1695	139	440	1167	1606	303	128	436	890	3424	225	591	1424	196	702	2629	5020	3989	1494	337	584	1282	212	744
4	399	2146	943	275	1053	144	440	780	1585	303	128	448	843	3578	214	585	1440	203	679	2759	5481	3961	1458	355	691	1324	211	744
5	381	2074	1415	292	1315	134	444	1119	1624	303	128	415	844	3578	214	585	1419	206	684	2593	5110	3969	1376	259	439	1209	211	728
6	425	2241	1449	312	3630	139	359	616	1520	330	128	459	863	3530	214	585	1445	193	694	2365	4593	3598	1221	321	524	1243	211	744
7	412	2174	733	234	1082	129	319	635	1328	330	138	428	822	3610	214	579	1378	203	678	2290	5534	3679	1264	321	393	1074	199	635
8	376	2057	1290	292	3186	144	447	853	1418	303	128	447	808	3530	225	585	1435	205	700	2215	5110	3804	1340	337	510	1007	199	587
9	367	1344	1160	275	1733	134	440	608	1049	279	120	418	910	3546	214	591	1383	203	684	1859	4538	3446	1717	336	555	1220	210	728
keskiarvo (Mt/min)	389	2050	1131	274	1766	138	419	889	1463	309	126	437	862	3528	216	586	1416	203	688	2416	5074	3789	1423	325	542	1201	207	694
keskihajonta (Mt/min)	18,9	273	251	26,9	1021	5,27	47	240	185	17,3	5,96	15	35,3	72,4	4,85	4	28,5	5,72	9,62	283	353	284	158	27,1	90,8	107	5,95	58,7

	8.1	8.2	9.1	10.1	10.2	10.3	11.1	12.1	13.1	13.2	13.3	13.4	14.1	15.1	16.1	16.2	16.3	16.4	16.5	17.1	18.1	19.1	19.2	19.3	19.4	19.5	20.1	21.1	22.1
1	2113	909	3085	345	51,6	269	1966	1697	975	332	137	474	355	741	305	45	59,7	290	120	1024	87,2	1506	3631	1822	233	633	411	3197	78,6
2	2243	921	3222	337	51,5	268	1995	1612	965	322	137	494	376	722	291	45,2	60,8	304	129	1164	101	1481	3256	1988	219	761	405	3453	78,1
3	2294	916	3146	329	51,5	276	1998	1639	936	313	137	481	367	755	293	44,2	70,7	298	154	1354	115	1464	3288	1996	219	640	433	3439	82,5
4	2269	928	3292	384	51,4	269	2031	1683	900	321	136	484	348	945	298	58,8	59,6	297	155	1033	125	2176	3777	1878	219	620	411	3389	76,4
5	2264	915	3187	371	50,5	283	2281	1739	964	327	136	453	309	731	296	46,9	59,6	295	139	1183	124	2054	3524	1903	219	627	437	3404	80,9
6	2326	933	3268	378	50,4	262	2144	1712	945	321	136	491	319	758	296	43,3	59,6	297	139	819	122	2091	3549	1857	219	608	420	3373	75,8
7	2248	927	3404	327	51,3	262	1882	1600	894	295	136	484	382	704	290	42,5	59,6	295	155	1394	133	1835	3491	1572	219	542	416	3369	80,1
8	1884	924	3042	363	51,3	251	1749	1469	902	312	128	474	320	615	290	43,8	59,6	291	153	1398	117	1890	3296	1593	219	528	407	3409	75,2
9	2015	931	3005	357	51,3	236	1622	1271	778	266	128	468	329	554	276	41,6	61,8	277	149	1186	87,9	1909	3533	1468	219	579	396	3136	75,1
keskiarvo (Mt/min)	2184	923	3183	355	51,2	264	1963	1602	918	312	135	478	345	725	293	45,7	61,2	294	144	1173	112	1823	3483	1786	221	615	415	3352	78,1
keskihajonta (Mt/min)	149	8,05	128	21,1	0,44	13,8	197	148	60,6	20,3	3,75	12,6	26,9	108	7,89	5,15	3,64	7,5	12,7	194	16,6	276	174	193	4,67	67,7	13,2	110	2,67

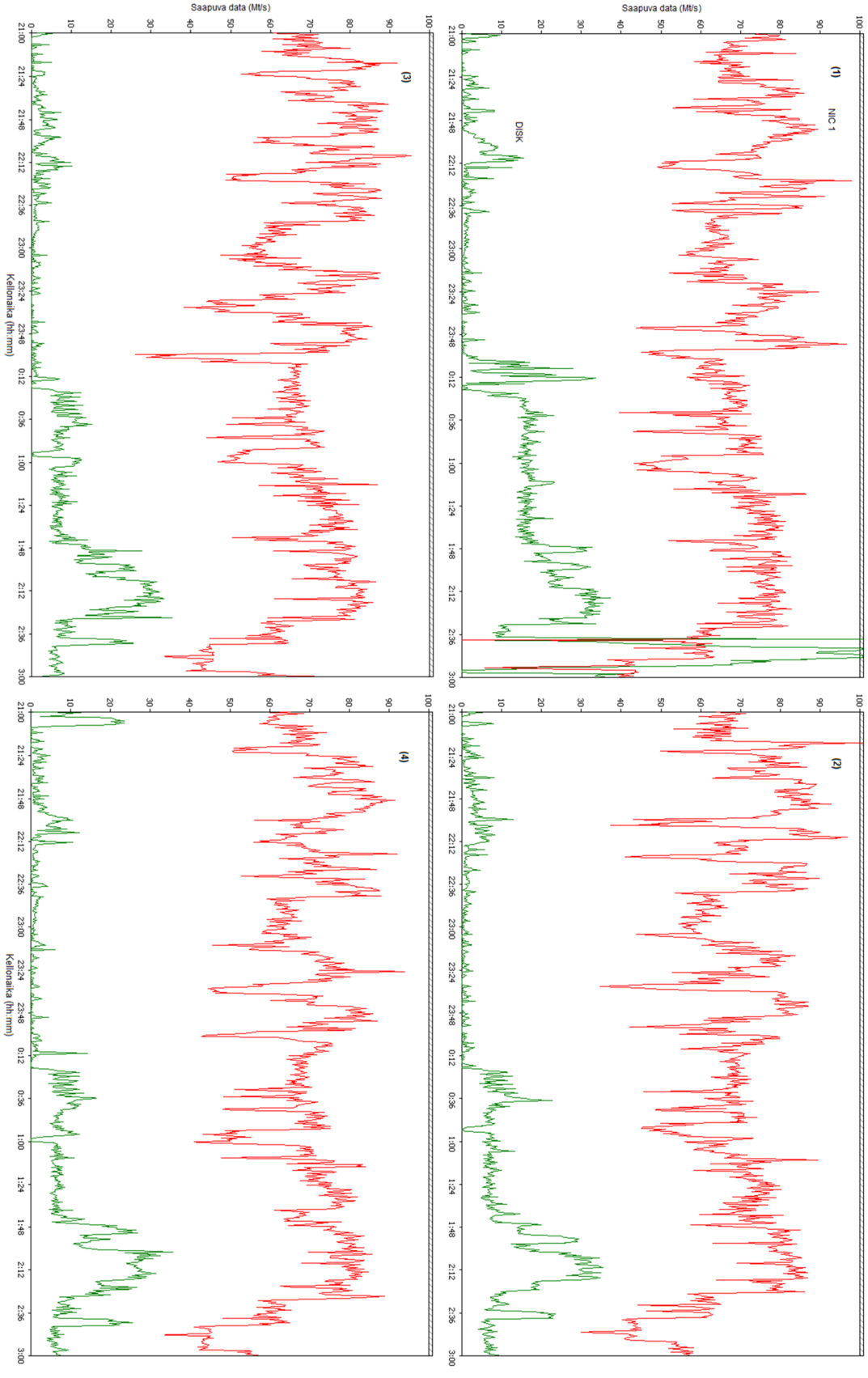
LIITE B: UUDEN JÄRJESTELMÄN VIIKOTTAISEN VARMISTUKSEN TILASTO

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	1391	1780	3331	1997	2293	2330	2440	2384	2288	1387	2313	2241	2640	2145	1565
2	1406	1736	3343	1980	2259	2388	2385	2529	2359	1471	1415	2389	2855	2310	1150
3	2376	1602	3274	2380	2062	2386	2348	2557	3177	1592	1383	2727	3213	2617	1841
4	2458	1797	3386	2634	2288	2488	2376	2665	3378	1654	1428	2215	2681	2340	3920
5	2011	1520	3102	2376	2147	2341	2352	2337	2239	1364	1206	2403	2375	2141	1535
6	2143	1580	3218	2564	2294	2411	2472	2621	2350	1407	1206	2469	2780	2226	1705
7	2436	1862	3343	2654	2346	2465	2521	2657	2668	1723	1411	3016	3276	2773	2053
8	2655	2961	2188	2364	2560	2864	2477	2429	2643	2660	2610	3266	1617	986	2541
9	2795	3233	2351	2563	2498	2837	2673	2497	1789	2524	2583	1924	1462	1605	3148
10	2837	3302	2383	2633	2605	2931	2692	2562	1771	1834	2732	3346	1820	1211	2527
Keskiarvo (Mt/min)	2251	2137	2992	2415	2335	2544	2474	2524	2466	1762	1829	2600	2472	2035	2199
Keskihajonta (Mt/min)	491,7	685,1	456,9	237,9	164,8	223,9	117,8	106,7	495,9	440,9	609,1	449,9	607,8	554,6	801,5

LIITE C: VARMISTUSPALVELIMEN PROSESSORIN JA MUISTIN KÄYTTÖASTE NELJÄNÄ PERÄTTÄISENÄ PÄIVÄNÄ



LIITE D: VARMISTUSPALVELIMELLE SAAPUVAN VERKKOLIIKENTEEN NOPEUS JA KIRJOITUSNOPEUS LEVYLLE NELJÄNÄ PERÄTTÄISENÄ PÄIVÄNÄ



LIITE E: IOMETER-SOVELLUKSEN ASETUKSET LUOTAESSA KEINOTEKOISTA LEVYKUORMITUSTA PALVELIMELLE